Journal of Machine Learning Research 27 (2026) 1-62 Submitted 5/24; Revised 12/25; Published 2/26

CHANI: Correlation-based Hawkes Aggregation of Neurons with bio-Inspiration

Sophie Jaffard JAFFARD@MPI-CBG.DE
Center for Systems Biology Dresden
Max Planck Institute of Molecular Cell Biology and Genetics (MPI-CBG)
Dresden, Germany

Samuel Vaiter SAMUEL.VAITER@UNIV-COTEDAZUR.FR
Laboratoire J. A. Dieudonné (LJAD)
CNRS, Université Côte d'Azur
Nice, France

Patricia Reynaud-Bouret PATRICIA.REYNAUD-BOURET@UNIV-COTEDAZUR.FR
Laboratoire J. A. Dieudonné (LJAD)
CNRS, Université Côte d'Azur
Nice, France

Editor: Jean-Philippe Vert

Abstract

The present work aims at proving mathematically that a neural network inspired by biology can learn a classification task thanks to local transformations only. In this purpose, we propose a spiking neural network named CHANI (Correlation-based Hawkes Aggregation of Neurons with bio-Inspiration), whose neurons activity is modeled by Hawkes processes. Synaptic weights are updated thanks to an expert aggregation algorithm, providing a local and simple learning rule. We were able to prove that our network can learn on average and asymptotically. Moreover, we demonstrated that it automatically produces neuronal assemblies in the sense that the network can encode several classes and that a same neuron in the intermediate layers might be activated by more than one class, and we provided numerical simulations on synthetic datasets. This theoretical approach contrasts with the traditional empirical validation of biologically inspired networks and paves the way for understanding how local learning rules enable neurons to form assemblies able to represent complex concepts.

Keywords: Hawkes process, Local learning rule, Expert aggregation, Spiking neural network, Neuronal synchronization

1. Introduction

Biological neurons and their organization into networks served as the inspiration for the perceptron (Rosenblatt, 1957), which pioneered artificial neural networks

(ANNs). This initial brain inspiration continued to drive the development of more sophisticated models such as the neocognitron (Fukushima and Miyake, 1982) and later convolutional neural networks (LeCun et al., 1989), all drawing from the biological processes observed in the visual cortex by Hubel and Wiesel (1962). Conversely, machine learning techniques can shed light on brain learning processes. Spiking neural networks (SNNs) (Tavanaei et al., 2019) illustrate well these interactions between the two domains: these networks model neurons activity by sequences of spikes representing the electrical pulses of biological neurons. More relevant than ANNs if one wants to study the brain neural code, they also provide insights into the development of energy-efficient algorithms (Stone, 2018). To be realistic, SNNs are trained thanks to local learning rules. However, there is a gap between the empirical results provided by these rules and the underlying theory.

In the present study, as an initial stride towards demonstrating mathematically that local learning rules enable neurons to form assemblies and induce global learning, we propose a network of spiking neurons called CHANI for Correlation-based Hawkes Aggregation of Neurons with bio-Inspiration. The task is as follows: the network should learn to classify objects into one of several classes by identifying relevant feature correlations, and its learning process should involve local transformations only. The model involves hidden and output nodes as postsynaptic neurons that produce spikes as a multivariate discrete-time Hawkes process (Ost and Reynaud-Bouret, 2020), whose spiking probability is a function of the weighted sum of the activity of presynaptic neurons at the previous time step. The learning algorithm used to update synaptic weights comes from the expert aggregation field (Cesa-Bianchi and Lugosi, 2006) thanks to this local paradigm: presynaptic neurons can be seen as experts and the strength of the connections between them and the postsynaptic neuron varies based on gains derived from these connections. Hidden neurons are trained to identify neuronal synchronization among presynaptic neurons, and a pruning process is employed to retain only those neurons that encode meaningful correlations, whereas output neurons are trained to respond to the classes in which the objects are classified.

Contributions. We present the first biologically inspired network which provably learns a classification task thanks to a local learning rule. Furthermore, our algorithm inherently generates neuronal assemblies: the network can encode multiple classes, and a single neuron in the intermediate layers may be activated by several classes.

• The main contributions of this article are summarized in Theorem 3. In a specific regime, referred to as CHANI EWA, we derive the explicit limit of the synaptic weights and show that CHANI converges to an idealized setting in which the hidden layers encode feature correlations. Moreover, after neuron selection, only the relevant feature correlations remain. When, in addition, classes are

defined by feature correlations, we prove that CHANI succeeds the learning task asymptotically.
• We establish several intermediate results leading to Theorem 3, as well as complementary results refining its conclusions. In particular, Theorems 9 and 14 provide rates of convergence for the synaptic weights. We also show that the network exhibits learning capabilities on average (Corollary 12), and we compute the VC-dimension of CHANI (Proposition 19). Finally, under more general assumptions, we derive regret bounds on the learning capabilities of hidden and output neurons (Propositions 22 and 24).
• We illustrate our theoretical analysis with numerical results on the classification of simulated data and handwritten digits (section 5).

Related work. In our prior work (Jaffard et al., 2024), we introduced a network named HAN (Hawkes Aggregation of Neurons) and established theoretical foundations for its learning capabilities. However, within the framework outlined in Jaffard et al. (2024), HAN had a very simple structure, lacking hidden layers, and could only achieve very simple tasks. In the present work, we extend these results to CHANI, which can support an arbitrary number of hidden layers designed to detect neuronal synchronization, and provide novel findings described above.

Our network is inspired by the cognitive model Component-Cue (Gluck and Bower, 1988), which states that an individual learns to classify objects by finding combinations of features which describe well the several object categories. The article Mezzadri et al. (2022) compared this model to another one called ALCOVE (Kruschke, 2020), which stipulates that people classify new objects by comparing them to previously learned ones, in order to see which one is closer to human behavior. They showed that often Component-Cue is a best fit to human learning. Note that Component-Cue is a classic ANN in the sense that is does not involve spiking neurons nor local learning rule, and it does not comprise hidden layers.

In the brain, conceptual objects are represented by analyzing and representing relationships among incoming signals. While elementary concepts can be depicted by the responses of individual neurons, more intricate ones are represented by groups of interconnected neurons that work together: we talk about neuronal assemblies (Singer et al., 1997). A single neuron can contribute to multiple assemblies, indicating that it may participate in representing different concepts. A current measure of assembly organization is based on correlations of firing among neurons. It has been shown on recorded and simulated data that this synchronicity can be caused by dynamic changes of synaptic connection strength (Gerstein et al., 1989), and neuroscientists have identified several local learning rules explaining these changes. As well-known example, Hebbian learning (Hebb, 2005) says that neurons that repeatedly fire together tend to become associated.

Spike-timing-dependent plasticity (STDP) (Caporale and Dan, 2008), a refined form of Hebbian learning, states that a connection is strengthened if the presynaptic neuron spiked just before the postsynaptic neuron, and weakened otherwise; this rule has been used in order to simulate neuronal assemblies (Litwin-Kumar and Doiron, 2014). However, there is no mathematical proof that these local rules enable to learn, nor that they produce neuronal assemblies. Establishing such theoretical guarantees would aid in comprehending how local mechanisms contribute to the formation of neuronal assemblies and how they function.

Enunciated in Legenstein et al. (2005), the Spiking Neuron Convergence Conjecture (SNCC) says that SNNs can learn to implement any achievable transformation. This conjecture is inspired by the perceptron convergence theorem (Rosenblatt et al., 1962), which asserts that as soon as the data to classify is linearly separable (*i.e.,* as soon as there exist weights with whom the data can be classified), then the weights given by the perceptron learning algorithm enable to correctly classify the data. This conjecture has been explored for networks using STDP as learning rule (Legenstein et al., 2005, 2008), with no conclusive theoretical result. In section 3.5.2, we prove a version of this conjecture in a very specific case for CHANI.

The multivariate Hawkes process (Hawkes, 1971) is a self-exciting and mutually exciting point process. Originally applied to the modelling of earthquake data (Türkyilmaz et al., 2013), its field of application is very broad: it is well-adapted to model networks of firing neurons (Hodara and Löcherbach, 2017), financial transactions (Hawkes, 2018; Bacry et al., 2015), health data (Bao et al., 2017), social networks (Zhou et al., 2013), or more generally, any sequences of events such that the occurrence of an event influences the probability of further events to occur. Also known as generalized linear models (GLMs) (Gerhard et al., 2017) in the literature, a lot of studies are about the estimation of its parameters (Reynaud-Bouret and Schbath, 2010; Kirchner, 2017), its mathematical properties (Brémaud and Massoulié, 1996) and its simulation in large networks (Bacry et al., 2017; Phi et al., 2020; Mascart et al., 2022, 2023). In neuroscience, it is used for instance to reconstruct functional connectivity (Reynaud-Bouret et al., 2013; Lambert et al., 2018), or to model networks of spiking neurons in order to study their mathematical properties (Galves and Löcherbach, 2016). However, a common assumption made when mathematically analyzing neural networks modeled by Hawkes processes is that their synaptic weights remain constant. This enables to achieve a stationary state, but prevents the modeling of learning behavior.

Hawkes processes have already been used to model events in a context of online learning: Chiang and Mohler (2020) proposes to solve spatio-temporal event forecasting and detection problems thanks to a multi-armed bandit algorithm, and Hall and Willett (2016) perform dynamic mirror descent to track how occurred events influence future events. However, none of these works use online learning algorithms to update the parameters of the Hawkes processes as we propose.

In the recent years, the emergence of transformers (Vaswani et al., 2017; Bietti et al., 2023) have revolutionized the field of deep learning, especially in natural language processing, but not only: for instance, the Transformer Hawkes Process (Zuo et al., 2020) incorporates attention modules into the formula of the conditional intensity of the Hawkes process in order to learn event sequence data. Many works have been done to understand them better: for instance, Rohekar et al. (2023) provide a causal interpretation of self-attention, and Cordonnier et al. (2020) investigate the relationship between self-attention and convolutional layers. In section 2.8, we draw an analogy between transformers and our model to contribute in understanding the remarkable effectiveness of transformers.

2. Presentation of the network learning algorithm

The structure and learning process of CHANI involve heavy notation that cannot be simplified without affecting the proofs. To improve readability, we alternate formal sections with "Intuition and Sketch" boxes, so that readers can focus on the main ideas. Additionally, Figure 1 gives the main steps of the algorithm.

For a quantity x_e indexed by $e \in E$, we use the following notations: $x_E = (x_e)_{e\in E}$ and $\langle x_e\rangle_{e\in E} = \frac{1}{|E|}\sum_{e\in E} x_e$. These notations are also used when the index appears as a superscript. The set of probability distributions over a set E is denoted $\mathcal{P}_E$. For all $n \in \mathbb{N}^*$, we denote $[n] = \{1, \dots, n\}$. All the other notations are listed in Table 3 in Appendix A.

2.1 Notation relative to the objects presentation during learning

We consider a supervised classification setting where the goal is to assign objects to classes $k \in K$ using a Spiking Neural Network (SNN).

More precisely, the SNN learns through sequential exposure to objects and is trained layer by layer. The l^{th} hidden layer, for $l = 1, \dots, L$, is trained during round l, which consists of the sequential presentation of M objects. Once all hidden layers have been trained, the output layer is trained during round $L + 1$ through the presentation of N objects.

Throughout the paper, and for any round —including the one involving N objects— each object is associated with an index m, which ranges from 1 to M for the rounds corresponding to the hidden layers ($l = 1, \dots, L$) and from 1 to N for the training of the output layer (round $L + 1$).

Each object m in every round is presented for a duration $T \in \mathbb{N}^*$.

Intuition & Sketch 1 (Time scales) *Figure 1.***A** *provides a schematic illustration of the successive presentation rounds. There are two time scales: M or N, representing the learning time and indexed by the presented object*

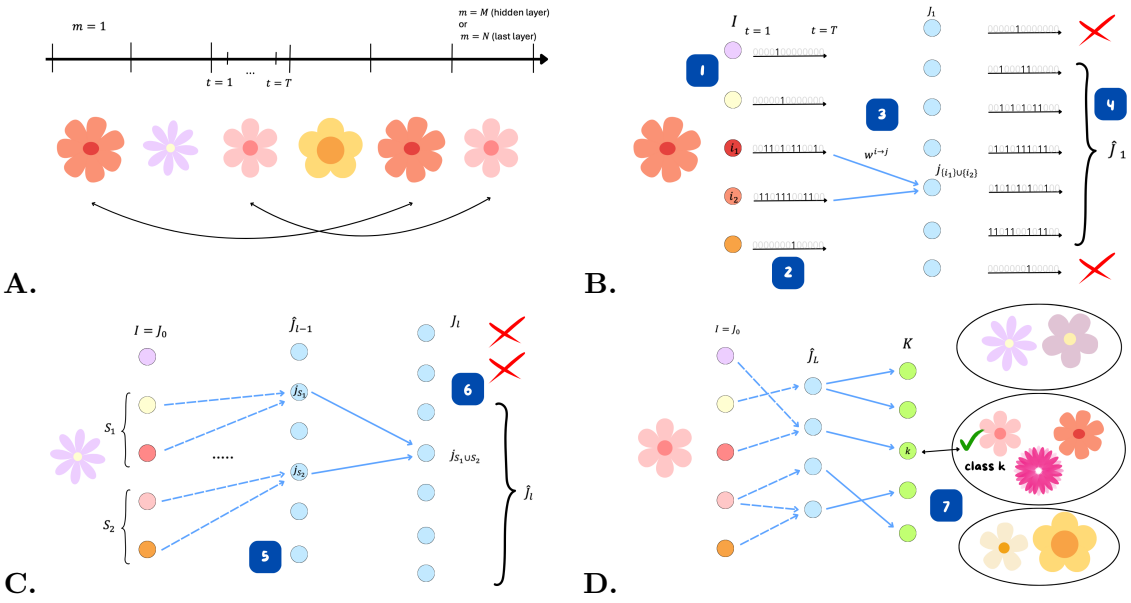

Figure 1: Sketch of CHANI algorithm. **A**. *CHANI's training consists of the sequential presentation of objects, with possible redundancy: here, objects 1 and 5 (respectively 3 and 6) share the same nature. Each object $m = 1, \dots, M$ or N (depending on whether hidden layers are being trained or not) is presented for a duration T.* **B**. *(Step 1) The network senses the presented object. (Step 2) Depending on the presence of a given feature i (e.g., color) in the object, neuron $i \in I$ emits more spikes during the presentation time T (i.e., more "1"s in the sequence of length T that it produces). (Step 3) In the first hidden layer, each neuron $j_{\{i_1\}\cup\{i_2\}} \in J_1$ searches for correlations between features i_1 and i_2 in the presented objects and updates its synaptic weights $w_m^{i\to j}$ after processing the m^{th} object. Ideally, at the end of training, it spikes only when the object contains both features i_1 and i_2. (Step 4) Neurons that do not detect sufficient correlation across all presented objects during the first training phase (presentation of M objects) are pruned. The resulting layer is denoted $\hat{J}_1$.* **C**. *(Step 5) As the layers are sequentially trained, a neuron j_S in layer J_{l-1} learns to detect the presence of all features $i \in S \subset I$ in the presented object. Ideally, at the end of training, this neuron spikes only when the object exhibits all features in S. The dashed arrows represent composite connections across several layers. (Step 6) In layer J_l, a neuron $j_{S_1\cup S_2}$ searches for correlations between neurons j_{S_1} and $j_{S_2} \in \hat{J}_{l-1}$. Ideally, after learning, it spikes only when the presented object contains all features in $S_1 \cup S_2$. All layers are successively pruned to retain only the relevant correlations in the presented objects.* **D**. *(Step 7) Once all hidden layers have been trained, the final layer K is added. Each neuron $k \in K$ identifies the neurons $j_S \in \hat{J}_L$ that are the most sensible to objects of class k. At the end of training, ideally, neuron k is connected to all $j_S \in \hat{J}_L$ such that the feature set S is relevant for defining class k.*

m; and a more microscopic scale, T, which corresponds to the time allowed for the network to spike and is indexed by time $t = 1, \ldots, T$.

2.2 The input layer

Intuition & Sketch 2 (Sensing of the object) *When an object is presented to the network for a duration T, the input neurons sense the object and start to spike (see Figure 1.***B***, Step 1). These input neurons are tuned to detect features (see Figure 1.***B***, Step 2). Depending on the presence or absence of features —or possibly on the degree to which a feature is present— the firing rate of these input neurons changes accordingly. As a first approximation, one can assume that neuron i remains silent if feature i is absent, and that it spikes at a fixed rate if the feature is present. However, our network can also handle residual firing rates when a feature is absent, or firing rates that vary with the degree of presence of a given feature. For instance, in the numerical results on classification of handwritten digits (Section 5.2), a feature corresponds to a pixel, and the firing rate is proportional to its darkness.*

Neurons of the input layer are always indexed by a i. A neuron of the input layer $i \in I$ produces a sequence of i.i.d. Bernoulli variables $X_{m,t}^{i,l}$, for $t = 1, ..., T$ during the presentation of object $m = 1, ..., M$ or N in round $l = 1, ..., L+1$. This activity is independent of the previous activities of all the neurons of the input layer of the same learning session, *i.e.,* $X_{m,t}^{i,l}$ is independent from $X_{m,s}^{i',l}$ for $i' \in I$ and $s < t$. However we do not assume that $X_{m,t}^{i',l}$ is independent from $X_{m,t}^{i,l}$ for $i \neq i'$.

Intuition & Sketch 3 (Nature versus object) *To ensure consistency and enable theoretical results, we need to establish a link between the firing distribution of the input neurons and the presented object. For this reason, we introduce a second concept: the nature of an object. The nature of an object refers to its intrinsic characteristics that are impacting the input layer. In particular, the input neurons respond in the same way (in a distributional sense) for all objects sharing that nature. The simplest way to think about this is to imagine that some objects represent exactly the same entity. For instance, in Figure 1.***A***, the same flower is presented multiple times. If these different occurrences are considered as distinct objects (with different indices m), they share the same nature, and the firing rates of the input neurons should therefore be identical. The nature of an object is indexed*

by $o \in \mathcal{O}$. The only thing the network can learn is a mapping from the set of natures $\mathcal{O}$ to the set of classes K, since only the nature of the object influences the activity of the input neurons.

Let o_m^l and $o_{m'}^{l'} \in \mathcal{O}$ be the natures of two objects m and m' presented respectively during rounds l and l'. If $o_m^l = o_{m'}^{l'} = o$ then since the distribution of the input neurons is **nature-only** dependent, we have that

$$\forall t, s = 1, ..., T, \forall i \in I, \mathbb{P}(X_{m,t}^{i,l} = 1) = \mathbb{P}(X_{m',s}^{i,l'} = 1).$$

We call this probability p_o^i, this is the firing rate of neuron i when exposed to an object of nature o.

Note that the objects o_m^l may be interpreted as training samples; however, we retain the term object to emphasize that, in our setting, the training samples have a particular structure. A training object o_m^l will influence the input neurons in the same manner as a test object of the same nature once learning has taken place.

Intuition & Sketch 4 (About the features) *Remark that we use the same notation for the index $i \in I$ of a neuron in the input layer, and the feature $i \in I$ that this neuron senses. If we follow this logic, a feature $i \in I$ is the characteristic of the object that is modifying the firing rate of neuron i. For instance, it can be the presence/absence of a certain color, and here the features $i \in I$ are colors, as well as the neuron $i \in I$ that detects the presence/absence of this color (see Figure 1.***B** *Step 1 and 2). It can also be the level of gray of pixels and here the feature $i \in I$ is a pixel as well as the neuron $i \in I$ that is sensitive to the pixel intensity (see Section 5.2).*

2.3 The hidden layers

A neuron of a hidden layer is always indexed by a j.

The hidden layers are constructed recursively, starting from the input layer, which is assimilated to the hidden layer J_0, with the correspondance that a neuron $i \in I$ can also be seen as a $j_{\{i\}} \in J_0$. In layer J_0, only singletons are allowed as subscripts. Then by recursion, for $l \geq 1$, the l^{th} hidden layer consists in neurons j_S, where $S \subset I$ and there exists $S_1 \neq S_2 \subset I$ such that j_{S_1} and j_{S_2} are in layer $l-1$. By recursion, it follows that a $j_S \in J_l$ corresponds to an S with cardinal $|S| = 2^l$.

Not all possible sets of cardinal 2^l are used. Indeed the layers are trained recursively and pruned. More precisely, to fix notation, let I_l be the set of all subsets of I with cardinal 2^l. If at the first hidden layer, J_1 can be put in one-to-one correspondence with I_2, only a subset $\hat{J}_1$ is kept at the end of the training phase, so

that J_2 corresponds only to union of pairs of sets appearing in $\hat{J}_1$. More generally, for all l, we only have $\hat{J}_l \subset J_l$ which can be put in one-to-one correspondence with a subset of I_l.

Intuition & Sketch 5 (About the indexation by a subset S) *In short, a neuron j_S in a hidden layer is associated with a subset $S \subset I$ and aims to be sensitive to all objects that "possess" all features $i \in S$. After training, ideally j_S spikes if and only if all neurons $i \in S$ are sufficiently active (see Figure 1.***C** *Step 5). If before the training of layer J_l a certain number of possible sets S are envisioned, the layer is pruned and becomes $\hat{J}_l$ to consider only the sets S that are really present in the data. For instance, if there is no blue and pink flower in the data that are presented to the network (see Figure 1.***B** *Step 4) and if the training went well, the neuron $j \in J_1$ that should detect the co-occurrence of pink and blue would not produce many spikes and would be removed before going to the training of layer 2. This pruning happens after each learning round l, which trains layer l (see Figure 1.***C** *Step 6).*

During learning round l, at the presentation of object m, the activity of a neuron j of an hidden layer $l' \leq l$, at time $t = 1, ..., T$, is given by $X^{j,l}_{m,t} \in \{0, 1\}$. It obeys a non-linear discrete Hawkes process, with random synaptic weights. More precisely, if we consider the filtration

$$\mathcal{F}^l_{m,t} = \sigma\left(X^{j,l'}_{m',s},\ m' \leq m,\ s \leq T \text{ or } s \leq t \text{ if } m' = m,\ j \in \bigcup_{l"=0,...,l'-1} \hat{J}_{l"} \cup J_{l'},\ l' \leq l \right),$$

which represents the σ-algebra generated by all the activity of all neurons, in layer less than l, during all the learning rounds $l' \leq l$ until the time t of the presentation of object m in round l, we have that, for a neuron $j \in J_l$,

$$p^{j,l}_{m,t}(w^j_m) := \mathbb{P}(X^{j,l}_{m,t} = 1 | \mathcal{F}^l_{m,t-1}) = (\psi^{j,l}_{m,t}(w^j_m))_+ \tag{1}$$

where

$$\psi^{j,l}_{m,t}(w^j_m) := -\nu + w^j_m \cdot X^{\hat{J}_{l-1},l}_{m,t-1}$$

where

• $w^j_m = (w^{j' \to j}_m)_{j' \in \hat{J}_{l-1}} \in \mathcal{P}_{\hat{J}_{l-1}}$, are the random presynaptic weights of neuron j, where $\mathcal{P}_{\hat{J}_{l-1}}$ is the set of probability vectors over the set $\hat{J}_{l-1}$. They are changing at each object m that is presented (see Section 2.5.3) and they are $\mathcal{F}^l_{m-1,T}$ measurable;

• $\cdot$ refers to the usual scalar product in $\mathbb{R}^{|\hat{J}_{l-1}|}$;

• the quantity $\nu \geq 0$ combined with $(\cdot)_+$ acts as a threshold, providing non-linearity to the network response (see Section 3.6 for a discussion about its role);

• with the convention $X^{j',l}_{m,0} = 0$.

To simplify the exposition, inhibition was not incorporated into the model, although the framework could be extended to include it without difficulty. Self-interactions were also omitted in order to facilitate the theoretical analysis

Moreover at the end of the learning phase l, the weights are frozen to their value w^j_{M+1} for all further use, so that the activity of a neuron $j \in \hat{J}_{l'}$ with $0 < l' < l$ during learning round l, is given by

$$p^{j,l}_{m,t}(w^j_{M+1}) := \mathbb{P}(X^{j,l}_{m,t} = 1 | \mathcal{F}^l_{m,t-1}) = (\psi^{j,l}_{m,t}(w^j_{M+1}))_+,$$

where the w^j_{M+1} have been computed during the learning round $l' < l$.

Intuition & Sketch 6 (About the presynaptic weights and ν) *The presynaptic weigths $w^{j' \to j}_m$, with $j' \in \hat{J}_{l-1}$ and $j \in J_l$ (see Figure 1.***B** *Step 3) are there to link the activities of the layer $l-1$ to the one of layer l. They evolve at each new object thanks to a local learning rule described later to mimic biological mecanisms (see Section 2.5.3), so that in particular during the learning phase, the activity of neuron j is not nature-only dependent. The quantity $\psi^{j,l}_{m,t}(w^j_m)$ can be seen as a very crude model of the membrane voltage of a neuron submitted to synaptic integration and in this sense, ν acts as a resting potential at which the neuron cannot spike. It is only if excited enough by its presynaptic neurons, that neuron j can spike.*

At the end of learning round l, and before starting learning round $l+1$, one presents each object $o \in \mathcal{O}$ only once. This is the selection phase. Therefore the activity of all neurons j in layer $l' \leq l$ is denoted $X^{j,l'}_{o,t} \in \{0,1\}$ and it obeys

$$\mathbb{P}(X^{j,l'}_{o,t} = 1 | \mathcal{F}^l_{M,T} \text{ and } X^{\hat{J}_{l'-1},l}_{o,1:t-1}) = (-\nu + w^j_{M+1} \cdot X^{\hat{J}_{l'-1},l}_{o,t-1})_+. \tag{2}$$

For every $l \in [L]$, a threshold $s_l \in (0,1]$ is fixed. At the end of the selection phase of layer l, the set of selected neurons is

$$\hat{J}_l := \{j \in J_l \text{ s.t. } \exists o \in \mathcal{O}, \langle X^{j,l}_{o,t} \rangle_{t \in [T]} \geq s_l\}. \tag{3}$$

In other words, the selected neurons $\hat{J}_l$ are the ones with an empirical spiking probability $\langle X^{j,l}_{o,t} \rangle_{t \in [T]}$ larger than the threshold s_l for at least one object of the selection phase. This corresponds to step 11 of Algorithm 1. Then for $l < L$, the set of neurons used to train the next layer is

$$J_{l+1} := \{j_{S_1 \cup S_2} \ : \ j_{S_1}, j_{S_2} \in \hat{J}_l \text{ such that } S_1 \cap S_2 = \emptyset\},$$

which is coding for feature subsets belonging to I_{l+1}. This selection encourages sparsity in the network in terms of the number of neurons.

Intuition & Sketch 7 (About the biological interpretation) *Our model is a caricature of some phenomenons in the brain. Indeed, if our learning rounds are very strict, it is true that during development, connections between different parts of the brain are done in a sequential fashion (Hevner, 2000). It is also true that during development and in particular, in the visual system (retina/ cortex), that some dummy signals, called retinal waves, are used to pretrained part of the system in particular in terms of correlations between presynaptic neurons and connections towards postsynaptic neurons (Feller and Kerschensteiner, 2020). Finally it is also true that a phenomenon called pruning happens at the synapse level so that the brain is progressively losing some neuronal connections and that this is an inherent part of the learning process (Lewis, 2011).*

2.4 The output layer

A neuron of a hidden layer is always indexed by a k and there is a one-to-one correspondence between the output neurons and the classes in which the objects are classified, so that we denote both by the same letter $k \in K$.

Once all the L hidden layers have been trained, we start the $L+1$ learning round, *i.e.,* we present N objects to the network to learn the final classification. During this round, at the presentation of object m, the activity of a neuron k is given by $X_{m,t}^{k,L+1} \in \{0,1\}$ and the activity of a neuron j of an hidden layer $l' \leq L$, at time $t = 1, ..., T$, is given by $X_{m,t}^{j,L+1} \in \{0,1\}$. The distribution of the hidden layers is as before, with weights fixed to w_{M+1}^{j}, whereas the distribution of $X_{m,t}^{k,L+1}$ is given by

$$p_{m,t}^{k}(w_m^k) := \mathbb{P}(X_{m,t}^{k,L+1} = 1 | \mathcal{F}_{m,t-1}^{L+1}) = w_m^k \cdot X_{m,t-1}^{\hat{J}_L,L+1}. \tag{4}$$

Again, the synaptic weights w_m^k form a probability distribution over the set of presynaptic neurons so $p_{m,t}^{k}(w_m^k) \in [0,1]$ a.s. At the end of the presentation of each objects, the weights w_m^k are updated and are $\mathcal{F}_{m-1,T}^{L+1}$ measurable (see Section 2.5.3). It corresponds to the Solo case of Jaffard et al. (2024). At the end of learning, the weights are frozen and we go back to a nature-only dependent behavior of the output, so that at the presentation of object with nature o, we have

$$\mathbb{P}(X_{o,t}^{k,L+1} = 1 | \mathcal{F}_{M,T}^{L+1} \text{ and } X_{o,1:t-1}^{\hat{J}_{l'-1},L+1}) = (-\nu + w_{M+1}^{j} \cdot X_{o,t-1}^{\hat{J}_{l'-1},l})_+. \tag{5}$$

The rule to classify objects is as follows: the object o_m is classified by the network in the class coded by the output neuron k which spiked the most during the presentation of the object.

Intuition & Sketch 8 (Classification rule and identity of neurons) *The output layer aims at describing the several classes in which the objects are classified. Ideally, after its training session, a neuron k would be active when an object belonging to class k is presented to the network, and non-active otherwise. The eventual activity of neuron k when presented with an object which does not belong to its class is considered as noise. Therefore, to correctly classify the object, the neuron coding for its class must overcome the noise coming from other neurons. However, since we are in a local learning rule set-up, neuron k does not have access, when modifying its weights, to the output of the other neurons $k' \neq k$. It has only access to its presynaptic activity as well as feedback from the environment, which is if the presented object belongs or not to class k (see Figure 1.***D*** (Step 7)). Biologically speaking, it is true that some neurons might have access, for instance by dopamine secretion, to at least a notion of success / failure (Frémaux and Gerstner, 2016; Schultz, 1998).*
More generally, the fact that we associate a class k and a neuron k or a set $S \subset I$ and a hidden neuron j_S is completely artificial. It is more likely that at random, many neurons get attributed to sets S or to class k with possible redundancy, so that in the end, at least most of the current classes k and set of features S are represented at least once. To avoid additional steps, we made the identifications and decided to have one and only one represent of class k and set S in the family of neurons.

2.5 Learning and local rules

2.5.1 Correlations

The hidden layers learning is based on correlation. We will need several notions of correlations. Let $I' \subset I$ a subset of features, $o \in \mathcal{O}$. Then the *correlation of I' w.r.t. object o* is

$$\rho_o(I') := \mathbb{P}\Big(\bigcap_{i \in I'} \{X^{i,1}_{o,T} = 1\} \Big),$$

i.e., the probability that the input neurons of the set I' spike together, the *average correlation* of I' is $\rho(I') := \langle \rho_o(I') \rangle_{o \in \mathcal{O}}$, *i.e.,* the average correlation of the input neurons of the set I' on every object and the *average correlation of I' w.r.t class $k \in K$* is $\rho^k(I') := \langle \rho_o(I') \rangle_{o \in k}$, *i.e.,* the average correlation of the input neurons of the set I' on objects of class k. Let $l \in [L]$, $m \in [M]$, J be a subset of neurons belonging to a layer with depth less than l. Then the *empirical correlation* of J when presented with object o^l_m is

$$\hat{\rho}^l_m(J) := \Big\langle \prod_{j \in J} X^{j,l}_{m,t} \Big\rangle_{t \in [T]},$$

which is an estimator of the probability that the neurons of the set J spike together when presented with object o_m^l.

2.5.2 Expert aggregation

The learning rules that are used to update the weights are based on expert aggregation (Cesa-Bianchi and Lugosi, 2006). Let us recall the main features and the corresponding notation. Here is a description of the expert aggregation problem. During M rounds, a forecaster can choose between several experts belonging to a set E. At each step m, each expert $e \in E$ has an unknown gain $g_m^e \in \mathbb{R}$. Thanks to the past knowledge, the forecaster chooses a probability distribution $p_m \in [0,1]^{|E|}$ over the set of experts E, and then receives the aggregated gain $g_m := p_m \cdot g_m^E$ where $g_m^E := (g_m^e)_{e \in E}$. This aggregated gain can be interpreted as the expectation of the gain that the forecaster would get by choosing one expert with probability p_m. Experts accumulate cumulated gains $G_m^E := (G_m^e)_{e \in E}$ were $G_m^e := \sum_{m'=1}^m g_{m'}^e$, as well as the forecaster which accumulates the cumulated gain $G_m := \sum_{m'=1}^m g_{m'}$. The *regret* of the forecaster measures how good its strategy is. It is defined as

$$R_M := \max_{q \in \mathcal{P}_E} \sum_{m=1}^M q \cdot g_m^E - G_m.$$

It compares the best possible cumulated gain with a constant strategy to the actual cumulated gain of the forecaster. It quantifies how close the strategy of the forecaster is to the optimal one.

An expert aggregation algorithm is a function $f : \mathbb{R} \times \mathbb{R}^{|E|} \mapsto [0,1]^{|E|}$ that the forecaster uses to update its strategy: the probability distribution chosen by the forecaster for the next round is

$$p_{m+1} := f(G_m, G_m^E).$$

Expert aggregation algorithms are designed to achieve small regret bounds, typically of the order of $\sqrt{M}$. Let us cite two main algorithms:

• EWA (Exponentially Weighted Average) determines the probability of selecting expert e at round m as

$$p_{m+1}^e = \frac{\exp(\eta G_m^e)}{\sum_{e' \in E} \exp(\eta G_m^{e'})}.$$

The parameter $\eta \geq 0$ is the learning rate. It quantifies how fast the forecaster learns.

• PWA (Polynomially Weighted Average) determines the probability of selecting expert e at round m as

$$p_{m+1}^e = \frac{(G_m^e - G_m)_+^{b-1}}{\sum_{e' \in E} (G_m^{e'} - G_m)_+^{b-1}},$$

where $b \geq 2$ is a parameter to choose.

These two expert aggregation algorithms employ different strategies: EWA ensures that each expert is assigned a strictly positive probability of selection, whereas PWA assigns a zero probability to experts whose cumulative gains are less than the forecaster's.

2.5.3 Weights update

Each neuron j of an hidden layer l learns by updating its synaptic weights $w_m^j = (w_m^{j' \to j})_{j' \in \hat{J}_{l-1}}$ during its learning round l by running its own expert aggregation algorithm f^l, indexed by l: the neuron j is the forecaster and the corresponding set of experts is the set of presynaptic neurons $\hat{J}_{l-1}$. A presynaptic neuron j' is attributed a gain $g_m^{j' \to j} \in \mathbb{R}$ for neuron j and the weights of j evolve according to the rule

$$w_{m+1}^j = f^l(G_m^j, (G_m^{j' \to j})_{j' \in \hat{J}_{l-1}})$$

where $G_m^j := \sum_{m'=1}^m w_{m'}^j \cdot g_{m'}^j$ is the cumulated gain of thepostsynaptic neuron j and $G_m^{j' \to j} := \sum_{m'=1}^m g_{m'}^{j' \to j}$ is the cumulated gain of the presynaptic neuron j'. This corresponds to steps 7 and 18 of Algorithm 1.

Most of the theoretical findings regarding expert aggregation algorithms apply to arbitrary bounded sequences of gains. Therefore, we can choose any bounded gains suitable for the learning task. For a hidden neuron $j_{S_1 \cup S_2} \in J_l$ where S_1 and S_2 are feature subsets encoded in the previous layers, the gain of presynaptic neuron j' is defined as

$$g_m^{j' \to j_{S_1 \cup S_2}} := \hat{\rho}_m^l(\{j_{S_1}, j_{S_2}, j'\}) \tag{6}$$

where $\hat{\rho}_m^l(\{j_{S_1}, j_{S_2}, j'\}) = \langle X_{m,t}^{j_{S_1},l} X_{m,t}^{j_{S_2},l} X_{m,t}^{j',l} \rangle_{t \in [T]} \in [0,1]$ is the empirical correlation between neurons j_{S_1}, j_{S_2} and j'. This choice of gain is designed to achieve the objective of neuron j training, which is to detect correlations between neurons j_{S_1} and j_{S_2}.

For an output neuron $k \in K$, the same set-up applies and we extend the gains defined in Jaffard et al. (2024). The gain of presynaptic neuron j is

$$g_m^{j \to k} := \begin{cases} \langle X_{m,t}^{j,L+1} \rangle_{t \in [T]} \times \frac{N}{N^k} & \text{if } o_m^{L+1} \in k \\ -\langle X_{m,t}^{j,L+1} \rangle_{t \in [T]} \times \frac{N}{N^{k'}} \times \frac{1}{|K|-1} & \text{if } o_m^{L+1} \in k' \neq k \end{cases} \tag{7}$$

where $N^{k'}$ is the number of objects belonging to class k' used to train the set of output neurons K. Therefore, when the presented object is in class k, *i.e.,* when neuron k should spike more than the other output neurons, it attributes positive gains to active presynaptic neurons. When the presented object is in another class, *i.e.,* when neuron k should stay silent, it attributes losses (negative gains) to active

presynaptic neurons. During the training phase of the output layer, the hidden layer L is already trained so the synaptic weights of a neuron $j \in \hat{J}_L$ are w^j_{M+1}.

Intuition & Sketch 9 (Local rules) *These choices of gains establish local rules: each neuron within a layer operates its own expert aggregation algorithm independently of the other neurons. The synaptic weights evolution of a neuron is solely influenced by the spikes of its presynaptic neurons. There are no backward passes, and the information regarding the true class of the presented object only plays a role in the output layer learning rule.*
For a neuron j in the hidden layers, the updates of the weights are solely based on correlations between presynaptic neurons. To some extent, it can be seen as a three-neurons pattern, trying to detect if one neuron responds in synergy with a couple of fixed neurons. If this rule is not close to any rules reported in the biological literature, up to our knowledge, it has the advantage to explain to some extent why synchronization of activity is so important in practice in the neuronal code (Singer et al., 1997).
By focusing on neural correlations and by applying the pruning process, the hidden layers are trained to learn feature correlations that are strongly expressed across the object natures in the set $\mathcal{O}$. Consequently, some of the learned correlations may not be directly relevant to the classification task, even if they are informative for characterizing the set $\mathcal{O}$. This limitation is unavoidable unless explicit class-related information is provided to the hidden layers.
An output neuron k also evolves thanks to a local learning rule that only depends on the presynaptic neurons behavior and is modulated by a reward factor. In this sense it looks close to the three-factors learning rule introduced by Frémaux and Gerstner (2016).
However note that, despite their local character, none of the rules we are applying are explicitly phrased as a gradient descent (no $\dot{w}$, derivative of w) and none of them take into account the postsynaptic firing rate per se, as it is usual in the biological learning rules model that have been studied in the literature.

2.6 The complete CHANI algorithm

The overall CHANI algorithm is summarized in Algorithm 1. Its complexity is determined by the number of calls made to the pseudo-random generator for obtaining Bernoulli variables. We need to simulate

$$(M+N)T(|I| + \max_{l=1,\dots,L} |\hat{J}_l|) + T \max_{l=1,\dots,L} |J_l|$$

random variables. Here the depth L is considered a constant.

Algorithm 1: CHANI

Initialization: $\hat{J}_0 := I$
for $l = 1$ ***to*** L **do**
  Initialization: $J_l := \{j_{S_1 \cup S_2} \ : \ j_{S_1}, j_{S_2} \in \hat{J}_{l-1}$ such that $S_1 \cap S_2 = \emptyset\}$, $\hat{J}_l := \emptyset$, $w^{j' \to j} := 1/|\hat{J}_{l-1}|$ for $j \in J_l$
  for $m = 1$ ***to*** M **do**
    Simulate $(X^{I,l}_{m,t})_{t\in[T]}$, $(X^{\hat{J}_1,l}_{m,t})_{t\in[T]}, \dots, (X^{\hat{J}_{l-1},l}_{m,t})_{t\in[T]}$ according to Section 2.2 and (1) .
    Compute gains $g^{J_l}_m$ according to (6).
    for $j \in J_l$ **do**
      $w^j_{m+1} \leftarrow f^l(G^j_m, (G^{j' \to j}_m)_{j' \in \hat{J}_{l-1}})$ `// aggregate experts`
  for $o \in \mathcal{O}$ **do**
    Simulate $(X^{I,l}_{o,t})_{t\in[T]}$, $(X^{\hat{J}_1,l}_{o,t})_{t\in[T]}, \dots, (X^{\hat{J}_{l-1},l}_{o,t})_{t\in[T]}$, $(X^{J_l,l}_{o,t})_{t\in[T]}$ according to Section 2.2 and (2).
    for $j \in J_l$ **do**
      if $\langle X^{j,l}_{o,t} \rangle_{t\in[T]} \geq s_l$ *and* $j \notin \hat{J}_l$ **then**
        add j to $\hat{J}_l$ `// select neuron` j
Initialization: $w^{j \to k} := 1/|\hat{J}_L|$ for $j \in \hat{J}_L$
for $m = 1$ ***to*** N **do**
  Simulate $(X^{I,L+1}_{m,t})_{t\in[T]}$, $(X^{\hat{J}_1,L+1}_{m,t})_{t\in[T]}, \dots, (X^{\hat{J}_L,L+1}_{m,t})_{t\in[T]}$ according to Section 2.4.
  Compute gains g^K_m according to (7).
  for $k \in K$ **do**
    $w^k_{m+1} \leftarrow f^{L+1}(G^k_m, (G^{j \to k}_m)_{j \in \hat{J}_L})$ `// aggregate experts`
Output: $\hat{J}_1, \dots, \hat{J}_L, (w^j_{M+1})_{j \in \hat{J}_1 \cup \dots \cup \hat{J}_L}, (w^k_{N+1})_{k \in K}$ `// selected neurons and final weights`

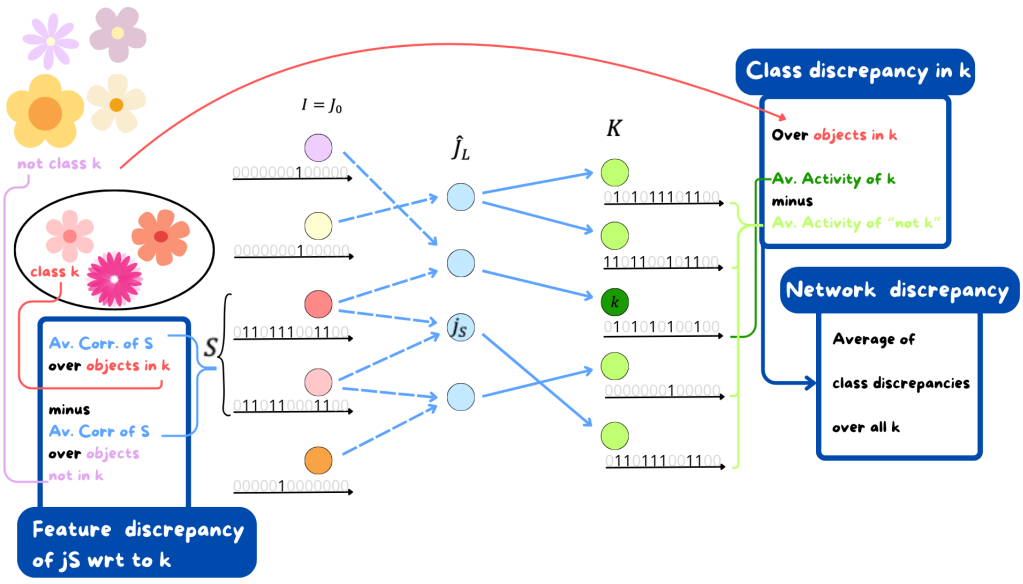

Figure 2: The three sorts of discrepancies. Av. stands for average.

2.7 Measure of the network performance

2.7.1 Spiking probability discrepancies

Since the classification rule of the network is based on who spikes the most, we measure CHANI's performance via discrepancies between firing rates or correlations. All these notions are represented in Figure 2.

First, we extend the notion of class discrepancy introduced in Jaffard et al. (2024), which compares the activity of an output neuron to the rest of the layer, as follows. We define *the average activity of neuron* k connected with weights q^k to the final hidden layer $\hat{J}_L$ during the presentation of object $m = 1, ..., N$ by

$$\hat{p}^k_m(q^k) = q^k \cdot \langle X^{\hat{J}_L, L+1}_{m,t} \rangle_{t=1,...,T}.$$

We want to know the performance of this choice of weights for class k. Hence, we introduce for a neuron $k \in K$ in the ouput layer, the *class discrepancy* of neuron k with weights $q^k \in \mathcal{P}_{\hat{J}_L}$:

$$\mathrm{Disc}^k_N(q^k, X^{\hat{J}_L, L+1}_{[N],[T]}) := \left\langle \hat{p}^k_m(q^k) - \hat{p}^{k'}_m(q^k) \right\rangle_{\substack{k' \,/\, k' \neq k \\ m \,/\, o^{L+1}_m \in k}} \in \mathbb{R}.$$

It compares the average activity of neuron k with the average activity of other neurons when presented with objects of class k. The larger the class discrepancy is, the more neuron k overcomes the noise in order to correctly classify the objects

of its class. It is a **global notion** since it involves the activity of all the output neurons.

The average class discrepancy of the network is called the *network discrepancy* with weights $q^K = (q^k)_{k\in K}$

$$\mathrm{Disc}_N(q^K, X^{\hat{J}_L,L+1}_{[N],[T]}) := \langle \mathrm{Disc}^k_N(q^k, X^{\hat{J}_L,L+1}_{[N],[T]})\rangle_{k\in K} \in \mathbb{R}. \tag{8}$$

The network discrepancy measures the average ability of the network to classify objects from all the classes.

To explain how the weights are strengthen, we need to find the sets S of features that are the most characteristic of a given class k. To do so, we extend the notion of feature discrepancy defined in Jaffard et al. (2024) for an input neuron coding for a feature to a neuron of the last hidden layer coding for a set of features. Let j be a neuron of the last hidden layer L, coding for a feature subset $S \in I^L$. The *feature discrepancy* of $j = j_S$ w.r.t. class $k \in K$ is the quantity

$$\mathrm{Disc}^{j_S\to k} := \rho^k(S) - \langle \rho^{k'}(S)\rangle_{k'\neq k} \in \mathbb{R}. \tag{9}$$

The feature discrepancy of neuron j_S quantifies how much the correlation of features of the set S is relevant to class k: it compares the average correlation of the input neurons coding for the features belonging to the set S w.r.t. class k with its average correlation w.r.t. other classes (see Figure 2 for an illustration). This new definition of feature discrepancy is much more complex than that of Jaffard et al. (2024). In our previous work, the feature discrepancy of an input neuron concerned only its individual activity, with no link to the activity of other input neurons whereas here it can capture complex patterns of input neurons activity.

2.7.2 Ideal network performance

In order to establish a performance target for CHANI, we define below ideal layers representing ideal performances.

Let us consider a network with conditional spiking probabilities given by (2) and (5), with synaptic weights given by a deterministic, fixed family q on the whole network. In this case, the distribution of a hidden or output neuron a is nature-only dependent and its activity when presented with an object of nature o, denoted $(X^a_{o,t})_{t\geq 1}$, is a sequence of Bernoulli variables. In particular, we denote

$$p^a_o(q^a) = \mathbb{P}(X^a_{o,T} = 1),$$

that is the *expected mean activity* of neuron a once all the dependencies with respect to the previous layers are taken into account (with $T > L + 1$).

Definition 1 (Ideal activities: layers and network)

• *For all $l \in [L]$, a neuron $j = j_S$ associated with the subset $S \subset I$ of the hidden layer l has an* **ideal activity** *with constant γ if*

$$\forall o \in \mathcal{O}, \quad p_o^j(q^j) = \gamma \rho_o(S).$$

A subset H_l of the hidden layer l with weight q^{H_l} is said to be **ideal** *with constant γ_l if all its neurons have an ideal activity with the same constant γ_l.*
• *A neuron $k \in K$ of the output layer has an* **ideal activity** *if*

$$\forall o \in \mathcal{O}, \quad p_o^k(q^k) > 0 \text{ if and only if } o \in k.$$

The output layer with weight q^K is said to be **ideal** *if all its neurons have an ideal activity.*
• *The network with weights q is said to be* **ideal** *when all its hidden and output neurons have an ideal activity.*

Intuition & Sketch 10 (About the ideal notion) *The previous notions formalize that in an ideal world,*
• *a hidden neuron $j = j_S$ encoding a subset S of features should be active when all input neurons $i \in S$ are active (*i.e., *when the presented object has all the features of the set S simultaneously), and its spiking probability should be proportional to the probability that they activate together.*
• *an output neuron k would code exactly for classes k without any noise.*
If being ideal might depend on the weights, the behavior of an ideal layer does not depend per se on the weights and a priori we do not know if there exist weight families for which our network is ideal.

To be able to compare more precisely our weights to the best possible weights we need therefore a measure that depends on the weights: the *ideal discrepancy*. It measures the performance of a fixed family of weights on the output layer $q^K \in (\mathcal{P}_{H_L})^{|K|}$, when connected to an ideal H_L, subset of the last layer, with constant 1:

$$\mathrm{Disc}^{\mathrm{id}}(q^K, H_L) := \langle \bar{p}_o^k(q^k, H_L) - \bar{p}_o^{k'}(q^{k'}, H_L) \rangle_{\substack{k \in K \\ o \in k \\ k' \neq k}} \tag{10}$$

with for every $k \in K$, $\bar{p}_o^k(q^k, H_L) := q^k \cdot (\rho_o(S(j)))_{j \in H_L}$, where $S(j)$ denotes the subset S of features such that $j = j_S$. This quantity measures the average difference between the spiking probability of an output neuron when presented with an object of its class and the other output neurons spiking probabilities: the larger it is, the better the performance of the output layer at classifying the objects. This notion is similar to the network discrepancy, but holds for an arbitrary output weight family connected to an ideal hidden layer.

If the output layer is connected to the ideal hidden layer H_L with an arbitrary constant γ_L, then by linearity the corresponding quantity is $\gamma_L \operatorname{Disc}^{\mathrm{id}}(q^K, H_L)$.

Note that in practice the last layer $\hat{J}_L$ is random. With H_L, we are considering a deterministic possible subset of neurons of the last layer, that can be put in a one-to-one map with a subset of I_L.

Definition 2 (Feasible weight family) *Let H_L be a set of neurons coding for feature subsets belonging to I_L. We say that $q^K \in (\mathcal{P}_{H_L})^{|K|}$, is*

- *a* **feasible weight family** *w.r.t H_L when*

$$\operatorname{Disc}^{\mathrm{id}}(q^K, H_L) > 0$$

We denote by $\mathcal{Q}_{H_L}$ the set of feasible weight families w.r.t. H_L.

- *a* **strong feasible weight family** *w.r.t. the set H_L when*

$$\forall k \in K, \forall o \in \mathcal{O}, \quad \bar{p}_o^k(q^k, H_L) > 0 \quad \textit{if, and only if,} \quad o \in k,$$

i.e., *when the output layer K, connected to the ideal layer H_L with constant 1 and with weights q^K, is ideal.*

Intuition & Sketch 11 (About the feasible weight families.) *The quantity $\bar{p}_o^k(q^k, H_L)$ is the expected mean activity of neuron k when connected with weights q^K to an ideal hidden layer with constant 1.*
Therefore the ideal discrepancy measures (in a deterministic fashion) the network discrepancy when the output layer is connected with weights q^K to an ideal layer. The fact that there exists feasible weight families means that if one is able to have an ideal last hidden layer, then one can find weights q^K to connect the last layer (encoding features correlations) to the classes in such a way that the resulting firing rates of the neurons lead to a good classification on average.
The notion of strong feasible weights, means that in the end, if one presents an object of class k not only the neuron k fires more than the other, but in fact it is the only one to fire.

Note that if a weight family q^K is a strong feasible family weight w.r.t. H_L then it is a feasible family weight w.r.t. H_L.

Another important remark is the following, which links all the notions of ideal activities and feasible weights. If we are given a family q of weights over the network and the last hidden layer H_L is deterministic such that

- H_L is ideal with constant γ_L
- the weights of the output layer q^K are strong feasible weights w.r.t. H_L

then the mean expected activity of an output neuron $k \in K$ satisfies

$$p_o^k(q^k) = \gamma_L \bar{p}_o^k(q^k, H_L)$$

and the output layer of the network is ideal.

2.8 Analogy with Transformers

The key innovation of the transformer architecture (Vaswani et al., 2017) is the self-attention mechanism, which allows the model to weigh the importance of different input tokens when processing each token in a sequence. This mechanism enables the model to capture long-range dependencies more effectively than traditional recurrent or convolutional neural networks. When using the specific expert aggregation EWA, CHANI shares similarities with transformers.

Input embedding. Objects to be classified have features which are embedded into point processes by the input layer.

Attention layers. Let $l \in [L]$, $j = j_{S_1 \cup S_2} \in J_l$ where S_1 and S_2 are feature subsets encoded in the previous layer, and $m \in [M]$. Then, the conditional spiking probability of neuron j during the presentation of object o_m^l of its learning phase reads

$$p_{m,t}^{j,l}(w_m^j) = \mathrm{softmax}(\eta^j A_{m-1}^j \cdot B_{m-1}) \cdot X_{m,t-1}^{\hat{J}_{l-1},l}$$

where $X_{m,t-1}^{\hat{J}_{l-1},l} = (X_{m,t-1}^{j',l})_{j' \in \hat{J}_{l-1}}$ is the activity of the previous layer at the previous time step of current object o_m^l, $A_{m-1}^j := (X_{m',t}^{j_{S_1},l} X_{m',t}^{j_{S_2},l})_{1 \leq m' \leq m-1, 1 \leq t \leq T}$ is the vector of correlations between j_{S_1} and j_{S_2} until previous object o_{m-1}^l, $B_{m-1} := (X_{m',t}^{j',l})_{j' \in \hat{J}_{l-1}, m' \in [m-1], t \in [T]}$ is the vector of previous layer activity until object o_{m-1}^l, and $A_{m-1}^j \cdot B_{m-1} := \left(\sum_{\substack{m' \in [m-1] \\ t \in [T]}} X_{m',t}^{j_{S_1},l} X_{m',t}^{j_{S_2},l} X_{m',t}^{j',l} \right)_{j' \in \hat{J}_{l-1}}$ is the matrix of correlations between j_{S_1}, j_{S_2} and neurons of the previous layer until previous object o_{m-1}^l. Therefore, the hidden layer of depth l acts as an attention layer enlightening correlations between neurons of the previous layers. At the end of the learning phase, the resulting weight matrix $w_{M+1}^{J_L}$ can be seen as a score matrix which is then used to select neurons coding for high correlations. The accumulation of hidden layers is analogous to a succession of attention layers.

Final linear layer. The output layer acts as a linear layer at the end of an attention module as the conditional spiking probability is linear in the activity of the last hidden layers.

3. Main theoretical results

In this section, we present the principal findings of the article by conducting a thorough analysis of the network's performance. We begin by stating the main theorem, followed by a series of intermediate theoretical results that both support its proof and offer insights extending beyond the theorem itself.

3.1 Main theorem

Let us define by recursion the sets $\bar{J}_l$ for all $l \in \{0, \dots, L\}$ as follows, with $s_1, \dots, s_l$ the thresholds defined in (3).
- $\bar{J}_0 := I$.
- For $l \geq 1$,

$$\bar{J}_l := \left\{ j_{S_1 \cup S_2} \;\middle|\; \begin{array}{l} j_{S_1}, j_{S_2} \in \bar{J}_{l-1} \text{ such that } S_1 \cap S_2 = \emptyset, \\ \text{and } \; \exists o \in \mathcal{O} : \; \rho_o(S_1 \cup S_2) > 2^{2^l - 1} s_l \end{array} \right\}.$$

For $l \geq 1$, the sets $\bar{J}_l$ are sets of neurons coding for high feature correlations for at least one object. They are the sets targeted during hidden layer pruning phases.

Consider the network defined with $\bar{J}_l$ as hidden layers, with fixed and deterministic weights defined by

$$\forall j' \in \bar{J}_{l-1} \text{ and } j = j_{S_1 \cup S_2} \in \bar{J}_l, \quad \bar{w}^{j' \to j} = \frac{1}{2} \mathbf{1}_{j' \in \{j_{S_1}, j_{S_2}\}}.$$

Define for the output layer, the subsets

$$\bar{J}_L^k = \arg\max_{j \in \bar{J}_L} \mathrm{Disc}^{j \to k}$$

that maximize the feature discrepancy and finish to construct the network with weights

$$\bar{w}^{j \to k} = \frac{1}{|\bar{J}_L^k|} \mathbf{1}_{j \in \bar{J}_L^k}.$$

Our main result consists in proving that this network is CHANI's limit under certain assumptions, several of which are detailed after the theorem because of their complexities.

Theorem 3 *Suppose we are under the CHANI EWA framework described in Section 3.2 and Assumptions 13 and 16 described in Sections 3.5.1 and 3.5.2 hold. In the regime where $N \to \infty$, $T/(NM^{L-1}) \to \infty$ and $N^{-1/2} e^{cM^{1/2}} \to \infty$ where C is a constant independent from M, N and T then, almost surely,*
- *each hidden layer converges: $\hat{J}_l \to \bar{J}_l$,*
- *the family of weights converges: $w \to \bar{w}$.*

Moreover, we have the following equivalence:
- *the limit output family weight is a strong feasible family w.r.t. $\bar{J}_L$*
- *there exists a strong feasible family w.r.t. $\bar{J}_L$*
- *each class $k \in K$ can be decomposed as unions of features correlations of size 2^L (Assumption 17).*

If this is the case, the limit network is ideal.

Intuition & Sketch 12 (About the limit network) *The main Theorem 3 shows the most refined result of the article: the* **limit network is ideal** *when classes are defined by combinations of feature correlations of size 2^L: more precisely, an object o belongs to class k if it possesses all the features of a given subset of size 2^L, or all the features of another such subset, or yet another, and so on. A neuron j_S in a hidden layer encodes exactly the correlations among the features in its subset S; that is, when presented with an object o, its spiking probability is proportional to $\rho_o(S)$. Similarly, a neuron k in the output layer encodes class k, meaning its spiking probability is strictly positive if and only if $o \in k$. Moreover, the output layer is entirely noise-free: an output neuron remains completely silent when the presented object belongs to a different class. While certain technical assumptions were required to establish this result, our numerical experiments in Section 5 indicate that CHANI performs well in practice even without all of these assumptions.*

Intuition & Sketch 13 (Spiking Neuron Convergence Conjecture) *Besides, thanks to the equivalence between the existence of a strong feasible weight family and the decomposition of each class as unions of feature correlations (Assumption 17 (class decomposition)), we know that* ***as soon as there exists a strong feasible weight family, the weights of the output layer of our network converge to one of these families with high probability.*** *We can see this statement as a version of the Spiking Neuron Convergence Conjecture (SNCC) (Legenstein et al., 2005) which says that spiking neural networks can learn to implement any achievable transformation: this conjecture is true for CHANI in this simplified framework. This result is unprecedented since, in our study Jaffard et al. (2024), we had to assume that the output limit weights were a feasible weight family in order to draw conclusions (see Corollary 3.5), without providing any criteria ensuring it.*

In the following sections, we provide detailed assumptions and theoretical results which ultimately lead to Theorem 3. In this purpose, we need to work within a precise framework that we call CHANI EWA, defined by the assumptions of the following section.

3.2 CHANI EWA Framework

Assumption 4 (EWA for hidden layers) *For any $l \in [L]$, the expert aggregation f^l used to train the hidden layer of depth l is EWA with learning rate $\eta^l = \sqrt{\frac{8\ln(|\hat{J}_{l-1}|)}{M}}$.*

Note that this choice of η^l maximizes the regret bound of EWA given in Cesa-Bianchi and Lugosi (2006). This assumption imposes no restriction on the expert aggregation algorithm f^{L+1} used to train the output layer.

Assumption 5 (Balanced) *During each training session of a layer, each nature of object $o \in \mathcal{O}$ is presented the same amount of times to the network.*

This is a technical assumption that we make to facilitate computations.

Assumption 6 (1/2 bias) *All hidden neurons have bias $\nu = \frac{1}{2}$.*

This assumption is here to ensure that hidden neurons activity converge to ideal hidden layers. See discussion about the choice of ν in Section 3.6.

Assumption 7 (Decreasing correlation) *For all $I' \subset I$ such that $\rho(I') > 0$, for all $i \in I \setminus I', \rho(I' \cup \{i\}) < \rho(I')$.*

This assumption means that adding a neuron to a set of correlated neurons results in a loss in correlation for at least one object. It is obviously verified when input neurons spike independently of one another. To ensure that input neurons activity verify this assumption, one can add a small independent perturbation to their activity.

Assumption 8 (Sparse features) *There exist thresholds $s_1, \dots, s_L \in (0, 1]$ such that the sets $(\bar{J}_l)_{0 \le l \le L}$ defined in section 3.1 verify*

$$\forall l \in \{0, \dots, L\} \text{ and } o \in \mathcal{O}, \quad |\{j_S \in \bar{J}_l, \rho_o(S) > 0\}| \le \frac{|\bar{J}_l|}{2}.$$

and

$$\forall l \in \{1, \dots, L\}, S \in I_l \text{ not encoded in } \bar{J}_l \text{ and } o \in \mathcal{O}, \quad \rho_o(S) < 2^{2^l - 1} s_l.$$

These thresholds are used for neuron selection (3).

This assumption states that there exist thresholds such that for any object $o \in \mathcal{O}$, at most half the neurons of I and $\bar{J}_l$ for $l \in [L]$ are active and spike together. Moreover, these thresholds strictly separate the neurons of the set $\bar{J}_l$ from other neurons during selection phases. This is a technical assumption that we need to study CHANI asymptotic behavior.

3.3 Hidden layers limit behavior: formation of neuronal assemblies

In order to study CHANI's performance, which is reflected by the activity of its ouput layer, we start by investigating the behavior of its hidden layers: in the framework CHANI EWA, we can explicitly compute the limit of hidden neurons weights.

Theorem 9 *Suppose we are in the CHANI EWA framework. Let $\alpha > 0$. There exist constants C_1 and C_2 independent of M and T and constants $c_1, c_2 > 0$ independent of M, T and the network parameters such that if $M \geq C_1$ and $T \geq C_2 M^{L-1}$, then with probability $1-\alpha$:*
(i) For all $l \in [L]$, $\hat{J}_l = \bar{J}_l$.
(ii) Let $l \in [L]$. Then, for all $j = j_{S_1 \cup S_2} \in \bar{J}_l$ where S_1 and S_2 are encoded in layer $\bar{J}_{l-1}$, at the end of learning the synaptic weights are such that

$$\left\| w^j_{M+1} - \frac{1}{2}\mathbb{1}_{\{j_{S_1}, j_{S_2}\}} \right\|_2 \leq c_1 \Big(\mathcal{J}_{l-1} \ln(\mathcal{J}_{l-1})^{\frac{1}{2}} \ln(L\mathcal{J}_{l-1}\alpha^{-1})^{\frac{1}{2}} \Big)^{l-1} \Big(\Big(\frac{M^{l-1}}{T} \Big)^{1/2} + e^{-c_2 \min_{l' \leq l-1} \left\{ 2^{-2^{l'}} \rho_{l'} \ln(|\bar{J}_{l'}|)^{1/2} \right\} M^{1/2}} \Big)$$

where the **limit weights** *are $\mathbb{1}_{\{j_{S_1}, j_{S_2}\}} := (\mathbb{1}_{j' \in \{j_{S_1}, j_{S_2}\}})_{j' \in \bar{J}_{l-1}} \in \{0,1\}^{|\bar{J}_{l-1}|}$ and where $\mathcal{J}_{l-1} := \max_{l' \leq l-1} |\bar{J}_{l'}|$, whereas $\rho_{l'} := \min_{j_S \in \bar{J}_{l'}} \rho(S)$ is the minimum average correlation of a feature subset of a hidden neuron of depth l'.*

This theorem states that the sets of interest $\bar{J}_l$ are selected with high probability. Besides, the weights of a hidden neuron $j = j_{S_1 \cup S_2}$ converge to uniform distribution on neurons j_{S_1} and j_{S_2} with an error term which is small when $M \gg 1$ and $T \gg M^{L-1}$. The dependency on the size of the selected sets $|J_l|$ also shows that the fewer neurons selected, the better. When taking only into account the dependency on M and T, the error term of the weights of layer l is in $O((\frac{M^{l-1}}{T})^{1/2} + e^{-C\sqrt{M}})$ where C is a constant independent of M and T. The exact error term is given in the proof of Theorem 9 (see Appendix C.3). It can be decomposed in two terms: the first one comes from the randomness induced by the spikes, as well as the accumulation of the errors coming from previous layers. The second one comes from the specific use of the expert aggregation algorithm EWA. This second term decreases as the average correlation of the feature subset of hidden neurons grows: the more hidden neurons describe high correlations between input neurons the better.

Intuition & Sketch 14 (About the convergence of hidden layers.) *With our assumptions, it means that the hidden neuron j_S meant to detect correlations of features in $S = S_1 \cup S_2$ finishes its learning by being linked*

only to j_{S_1} and j_{S_2}. It could have been wired like that in hard from the beginning and just the pruning part could have been kept. But this would have missed the biological interpretation point of view, which is that there is progressive learning of the correlations. An even more realistic point of view would have been, as quickly mentionned in Box 8, that a neuron j of an hidden layer picks at random two presynaptic neurons and focus on detecting the correlation of these two. Then we would need even more neurons as a first step in an hidden layer before pruning to detect all necessary connections. If the theoretical results are likely to be of the same flavor, this would have overcomplexified our procedure and we decided to not go down this path.
Note also that the gains of the presynaptic neurons used here (6) *are there to eventually reinforce link of j_S with other neurons j' that migh detect the same correlation as what j_S wants to do. In this sense, it might be looked as a classical "fire together, wire together" rule, ie the Hebbian rule which would focus only on correlation, and this is also this mecanism that makes the link with Transformers. However, with Assumption 7, we are in fact assuming that adding $j' \neq j_{S_1}$ and $\neq j_{S_2}$ will degrade the correlation strength, hence the fact that neuron j' is not reinforced through the local learning rule. It is possible that in a more complex network, with for instance loops, one can use additional connections to some j' to strengthen the answer of the post-synaptic neurons.*

Corollary 10 *Suppose we are in the CHANI EWA framework. When M tends to infinity with T/M^{L-1} tending to infinity as well, then almost surely*
- *For all $l \in [L]$, $\hat{J}_l \to \bar{J}_l$.*
- *For all $j = j_{S_1 \cup S_2} \in \bar{J}_l$, $w^j_{M+1} \to \frac{1}{2}\mathbb{1}_{\{j_{S_1}, j_{S_2}\}}$.*

Moreover, each of these limit hidden layers are ideal with constant $\gamma_l = 2^{-2^l+1}$, for $l \in [L]$.

Intuition & Sketch 15 (About neuronal assemblies.) *Combined with Theorem 9, this corollary ensures that at the limit the hidden layers of our network code exactly for feature correlations. A hidden neuron is activated when presented with objects having a certain combination of features (*i.e., *, having simultaneously all the features of a certain set). If this combination can be found in several classes, then the same neuron will be activated, and can therefore help to represent more than one class: in this sense, CHANI produces neuronal assemblies. These results, valid for any number of hidden layers, are unprecedented since in our previous work Jaffard et al. (2024) HAN did not have hidden layers.*

3.4 Average learning

Now that we studied the hidden layers behavior, we can analyze the network's performance in the classification task. In order to so, we compare the network discrepancy (8), which measures the average performance of the network during learning, to the largest ideal discrepancy of a feasible weight family (10), which represents an ideal performance. Let

$$E_{\text{hid}}(M,T,\alpha) := L\sqrt{\mathcal{J}_L} \max_{l\in[L],j\in\bar{J}_l} \left\| w_{M+1}^j - \bar{w}^j \right\|_2 \tag{11}$$

the maximum error term of a neuron of hidden layer, bounded in Theorem 9.

Note that if we have assumed EWA on the hidden layer (Assumption 4), one can use any expert agregation algorithm that they want on the last layer as long as the following assumption holds.

Assumption 11 (Output neurons regret bound) *The expert aggregation algorithm f^{L+1} used to train the output layer is such that for any set of experts E and any deterministic sequence of gains $g_{[N]}^E := (g_m^e)_{\substack{e\in E\\ m\in[N]}}$ taking value in $[a,b]$,*

$$R_M \leq C(b-a)\sqrt{\ln(|E|)M}$$

where C is a constant.

Note that both EWA and PWA verify this bound for certain parameters (see Cesa-Bianchi and Lugosi (2006) for details).

By combining Theorem 9 and Assumption 11 (Output neurons regret bound), we get the following result.

Corollary 12 *Suppose we are in the CHANI EWA framework and Assumption 11 (output neurons regret bound) holds. Let $\alpha > 0$. Suppose $M \geq C_1$ and $T \geq C_2 M^{L-1}$ where C_1 and C_2 are the constants defined in Theorem 9 and $\mathcal{Q}_{\bar{J}_L}$, the set of feasible weight families w.r.t. the ideal set of neurons $\bar{J}_L$, is non-empty. Then with probability $1-\alpha$ the conclusions of Theorem 9 hold and there exists a constant $c > 0$ independent of the network parameters and $\gamma_L = 2^{-2^L+1}$ such that*

$$\begin{aligned}\text{Disc}_N(w_{[N]}^K, X_{[N],[T]}^{\hat{J}_L,L+1}) \geq & \gamma_L \max_{q^K\in\mathcal{Q}_{\bar{J}_L}} \text{Disc}^{\text{id}}(q^K,\bar{J}_L) \\ & - c\left(\sqrt{\frac{|\mathcal{O}|}{NT}\ln\left(\frac{L|\bar{J}_L|}{\alpha}\right)} + |\mathcal{O}|\sqrt{\frac{\ln(|\bar{J}_L|)}{N}} + E_{hid}(M,T,\alpha)\right).\end{aligned}$$

This corollary establishes that CHANI's network discrepancy exceeds the ideal discrepancy of the best feasible weight family possible multiplied by a constant γ_L,

minus an error term which is small when $M \gg 1$, $T \gg M^{L-1}$ and $N \gg 1$, and increases with the number of selected neurons. This error term is threefold: the first part comes from the randomness of the spikes, the second one comes from the regret bound of the expert aggregation of the output layer f^{L+1}, and the third part comes from the approximation error between hidden layers weights and their limit. When taking only into account the dependency in T, M and N, the overall error term is in

$$O\Big(N^{-1/2} + \Big(\frac{M^{L-1}}{T}\Big)^{1/2} + e^{-C\sqrt{M}}\Big)$$

where $C > 0$ is a constant independent of M, N and T. Note that this corollary holds for any expert aggregation used to train the output layer, as long as it verifies Assumption 11 (output neurons regret bound).

In other words, on average, CHANI performs as well as it would if the output layer were connected to the ideal hidden layer $\bar{J}_L$ with constant γ_L with the best possible feasible weight family. This corollary compares CHANI's network discrepancy to a deterministic quantity which does not depend on CHANI itself, and which represents an ideal performance where the hidden layers detect neuronal synchronization and the output layer succeeds to rewrite the classes in terms of combination of correlations between 2^L features. In Section 4.2 Proposition 24, we give a weaker version of this result under more general assumptions, in which we compare CHANI's network discrepancy to the best network discrepancy using CHANI's own hidden layers.

Note that the error term increases with the depth L, and the constant γ_L decreases with L, which suggest that deeper networks give worse bounds. However, a major assumption is the existence of a feasible weight family. Since adding layers increases the complexity of the classes which can be represented by the network, this assumption is more likely to be verified for large L.

This corollary can be related to Theorem 3.3 of Jaffard et al. (2024), where HAN network discrepancy is compared to the best safety discrepancy of a feasible weight family, a quantity describing how well a feasible weight family classifies objects. However, this new result is much more general, for several reasons. Firstly, in the current work, CHANI can comprise any number of hidden layers, whereas in our previous work HAN is restricted to having solely an input and an output layer. Secondly, both of these results assume the existence of a feasible weight family. However, in (Jaffard et al., 2024), this assumption is considerably more restrictive, as a feasible weight family can only exist if the classes can be described as simple combinations of features. In our current study, a feasible weight family is achievable when classes can be characterized as combinations of feature correlations, the complexity of which grows with the depth L (see conditions for the existence of strong feasible weights families in section 3.5.2). In particular, if a feasible weight family does not already exist, the addition of layers may facilitate its emergence.

Intuition & Sketch 16 (Oracle inequality and global optimization) *This result can be seen as an oracle inequality in the sense that each individual neuron does not know what the other neurons in the network do since all rules are local. Still the performance of the global network is as good as (up to a multiple constant) the best choice of weight q^K on the output layer, coupled with ideal hidden layers that perfectly detect correlation. In this sense, it could also be compared to a global optimisation of the network discrepancy on the weights on the output layer. However it cannot be compared to a global optimization of the network discrepancy with respect to the weights on all the layers, because nothing says that another choice of weights for the hidden layers would not lead to a better discrepancy, in particular if the hidden layer do not encode correlations of features anymore.*

3.5 Whole network limit behavior

3.5.1 Output layer limit behavior

In this section, we work in the framework CHANI EWA, with the additional assumption that the expert aggregation used for the output layer is EWA as well (which is a stronger assumption than Assumption 11 (Output neurons regret bound)). This allows us to end the asymptotic study started in section 3.3 by computing the limit output weights.

Assumption 13 (EWA for the output layer) *The expert aggregation f^{L+1} used to train the output layer is EWA with learning rate $\eta^{L+1} = \frac{1}{|\mathcal{O}|}\sqrt{\frac{2\ln(|\bar{J}_L|)}{N}}$.*

Note again that this choice of η^{L+1} minimizes the regret bound of EWA given in Cesa-Bianchi and Lugosi (2006). For $\alpha \in (0,1]$ let

$$E_{\text{out}}(T,\alpha) := \sqrt{\frac{\ln(|\bar{J}_L|)}{|\mathcal{O}|T}\ln\Big(\frac{L|\bar{J}_L||K|}{\alpha}\Big)}$$

We recall that the feature discrepancy $\text{Disc}^{j\to k}$ of a hidden neuron $j = j_S$ of the last hidden layer w.r.t. an output neuron k quantifies how much the feature subset S is relevant to describe class k (9).

Theorem 14 *Suppose we are in the CHANI EWA framework and Assumption 13 holds. Let $\alpha \in (0,1]$. Suppose $M \geq C_1$ and $T \geq M^{L-1}C_2$ where C_1 and C_2 are the constants defined in Theorem 9. We recall that $\bar{J}_L^k = \arg\max_{j\in\bar{J}_L} \text{Disc}^{j\to k}$ and $\mathbb{1}_{\bar{J}_L^k} = (\mathbb{1}_{j\in\bar{J}_L^k})_{j\in\bar{J}_L}$. Then there exist constants $c_3, c_4 > 0$ independent of the*

network parameters, such that with probability $1-\alpha$*, the conclusions of Theorem 9 hold and for all* $k \in K$

- *if* $\bar{J}_L^k = \bar{J}_L$ *then*

$$\left\| w_{N+1}^k - \frac{1}{|\bar{J}_L^k|} \mathbb{1}_{\bar{J}_L^k} \right\|_2 \leq c_3 \Big(\sqrt{|\bar{J}_L| \ln(|\bar{J}_L|) N} E_{hid}(M,T,\alpha) + E_{out}(T,\alpha) \Big).$$

- *if* $\bar{J}_L^k \subsetneq \bar{J}_L$ *let* $\Delta^k := \max_{j \in \bar{J}_L^k} \mathrm{Disc}^{j \to k} - \max_{j \in \bar{J}_L \setminus \bar{J}_L^k} \mathrm{Disc}^{j \to k}$*. Then*

$$\left\| w_{N+1}^k - \frac{1}{|\bar{J}_L^k|} \mathbb{1}_{\bar{J}_L^k} \right\|_2 \leq c_3 \Big(\sqrt{|\bar{J}_L| \ln(|\bar{J}_L|) N} E_{hid}(M,T,\alpha) + E_{out}(T,\alpha) + e^{-c_4 \frac{\Delta^k}{2^{2L}} \sqrt{\ln(|\bar{J}_L|) N}} \Big).$$

Note that the bound given by Theorem 9 on the error term $E_{\mathrm{hid}}(M,T,\alpha)$ (11) is independent of the size of the last hidden layer $|\bar{J}_L|$ and the number of presented objects for the training phase of the output layer N. The overall error is threefold: the first two parts come from the approximation error between hidden and output layer weights and their limit, and the third part comes from the use of the expert aggregation EWA for the output layer. When taking only into account the dependency on M, N and T, the error is in

$$O\Big(\Big(\frac{NM^{L-1}}{T} \Big)^{1/2} + N^{1/2} e^{-CM^{1/2}} + e^{-DN^{1/2}} \Big)$$

where C and D are constants independent of M, N and T: it is small when $N \gg 1$, $M \gg \ln(N)^2$, $T \gg NM^{L-1}$ and increases with the number of selected neurons.

Corollary 15 *Suppose we are in the CHANI EWA framework and Assumption 13 holds. When* T*,* M *and* N *tend to* ∞ *with* $T/(NM^{L-1})$ *and* $e^{CM^{1/2}} N^{-1/2}$*, where* C *is a constant independent of* M, N *and* T*, tending to infinity as well, then almost surely, the conclusions of Corollary 10 hold and for all* $k \in K$*,*

$$w_{N+1}^k \to \frac{1}{|\bar{J}_L^k|} \mathbb{1}_{\bar{J}_L^k}.$$

This corollary states that the weights of an output neuron k converge to the family which uniformly distributes weights on presynaptic neurons with maximal feature discrepancy. Therefore, at the limit, neuron k will be connected only to neurons coding for correlations which are sensible to class k.

Theorem 14 and Corollary 15 can be related to Theorem 3.4 of Jaffard et al. (2024), where we computed HAN limit weights with no hidden layers. Here, we generalize this asymptotic analysis to CHANI. Besides, thanks to the addition of hidden layers, CHANI's limit output weight family is more interesting than HAN's

from a learning point of view since it decomposes classes in terms of correlations of 2^L features (against combinations of simple features). In a general case, we cannot say if this limit family is a strong feasible weight family, or even if it enables the network to correctly classify objects when noise is allowed. However, we can conclude in a more precise framework defined below where classes are defined by feature correlations.

3.5.2 When classes are defined by feature correlations

First, let us define some assumptions that we will need to perform our study.

Assumption 16 (Binary correlations) *There exists $p \in (0,1]$ such that for all $j_S \in \bar{J}_L$, for all $o \in \mathcal{O}$, $\rho_o(S) \in \{0, p\}$. If $\rho_o(S) = p$ then we say that object o has features S. Besides, for $j_S \in \bar{J}_L$, let $\mathcal{O}^S := \{o \in \mathcal{O}, \rho_o(S) = p\}$ the set of objects with features S. Then all the $\mathcal{O}^S$ have the same cardinality.*

The first part of this assumption is verified if for instance the input neurons spike independently of each other and have a fixed spiking probability when they are activated. The restriction on the cardinality of the sets $\mathcal{O}^S$ is a technical assumption that we make in order to facilitate the computations.

Assumption 17 (Class decomposition) *For every $k \in K$, there exists a set $E^k \subset I_L$ such that for every $S \in E^k$, $j_S \in \bar{J}_L$ and $k = \bigcup_{S \in E^k} \mathcal{O}^S$.*

This assumption means that the classes in which the objects are classified are defined by feature correlations (encoded in the set $\bar{J}_L$). To be more precise, an object o belongs to class k if it possesses all the features of a given subset of size 2^L, or all the features of another such subset, or yet another, and so on. Thus, the complexity of these correlations increases with L: as the network's depth grows, a set $\mathcal{O}^S$ captures more intricate patterns within the objects.

Theorem 18 *Suppose Assumption 16 (binary correlations) holds. Then the following statements are equivalent.*
- *Assumption 17 (class decomposition) is verified.*
- *There exists a strong feasible weight family w.r.t. the set $\bar{J}_L$.*
- *The limit output weights family defined in Corollaries 10 and 15 is a strong feasible weight family.*

In particular, when Assumption 17 (class decomposition) is verified, the network with limit weights is ideal.

This final theorem provides the last ingredients needed to establish the main result of our article, which is Theorem 3. It establishes that if the classes can be characterized by correlations among 2^L features encoded in the input layer,

then CHANI is capable of successfully learning to classify them. Furthermore, as long as there exists an output weight family that enables correct classification when the hidden layers encode correlations, CHANI's output layer will converge to one of these optimal weight families. This learning guarantee is analogous to the Perceptron Learning Theorem (Rosenblatt et al., 1962), which states that if a set of weights exists that can classify the data—equivalent to assuming linear separability—then the perceptron learning algorithm will find such weights. Additionally, this result can be linked to the Spiking Neuron Convergence Conjecture that we described in Intuition & Sketch 13: we proved this conjecture for CHANI within a specific framework.

In this framework, we can also compute the Vapnik-Chervonenkis dimension of our network. For a given classification algorithm, its VC-dimension corresponds to the size of the largest set of points that it can shatter. It measures the complexity of the class of functions that can be learned. In the literature, this dimension appears in generalization error bounds.

We assimilate the objects to be classified as sequences $o = (o_i)_{i \in I}$ where $o_i = 1$ if o has feature i and $o_i = 0$ otherwise: they live in a space of dimension $|I|$. Neuron $j_S \in \bar{J}_L$ is activated when o has all the features of the set S. We consider a binary output: let the hypothesis set $H := \{\mathbb{1}_F \text{ such that } F \subset J_L\}$, where for an object o, $\mathbb{1}_F(o) = 1$ if and only if there exists $j \in F$ activated by o, and 0 otherwise. It corresponds to the functions that CHANI can learn. Then we have the following result.

Proposition 19 *The VC-dimension of the hypothesis set H is the size of the last hidden layer*

$$\mathrm{VCdim}(H) = |\bar{J}_L|.$$

Unlike multilayer perceptrons, whose VC-dimension scales as $O(W \log W)$ (where W denotes the total number of weights) (Shalev-Shwartz and Ben-David, 2014), CHANI's VC- dimension depends solely on the complexity of the last hidden layer. This arises because each hidden neuron autonomously learns correlations in an almost unsupervised learning manner, as its learning rule does not account for the true class of the presented object. Besides, the complexity of one hidden layer is not added to the next, but rather included in it. However, as the number of selected neurons increases, the errors in the results regarding the asymptotic behavior of CHANI also increase. Therefore, it is essential to strike a balance between accurately representing complex classes and learning efficiently.

3.6 Discussion about the choice of bias.

Assumption 6 (1/2 bias) could be relaxed to any $\nu \in [1/2, 1)$: Theorem 9 and Corollary 10 would still hold. Indeed, their key ingredient is that hidden neurons spiking probabilities converge to ideal spiking probabilities. The conditional

spiking probability of hidden neuron $j = j_{S_1 \cup S_2}$ with bias ν and fixed weights $\mathbb{1}_{\{j_{S_1}, j_{S_2}\}}$ where S_1 and S_2 are feature subsets encoded in the previous layer (which are the limit weights given by Corollary 10) at time step t of the presentation of an object o is $\Big(-\nu + \frac{1}{2}(X_{o,t-1}^{j_{S_1}} + X_{o,t-1}^{j_{S_2}}) \Big)_+$, where $X_{o,t-1}^{j_{S_1}}$ and $X_{o,t-1}^{j_{S_2}}$ are the activities of neurons j_{S_1} and j_{S_2} at the previous time step. For $\nu \in [1/2, 1)$, neuron j is active if and only if neurons j_{S_1} and j_{S_2} spike, and in this case its conditional spiking probability is $1-\nu$. Therefore, the limit spiking probabilities would still be ideal. However, there is a decrease in firing rate from one hidden layer to the next and the choice $1/2$ guarantees the lowest decrease. This decrease appears in the constant γ_L of Corollary 12: in fact, $\gamma_L = (1-\nu)^{2^L - 1}$ and a large γ_L guarantees a large class discrepancy for our network. Therefore, the choice $\nu = 1/2$ is optimal.

For $\nu < 1/2$, neuron j is active even when only one of the two neurons j_{S_1} and j_{S_2} spikes, so this choice of bias does not provide ideal limit spiking probabilities and the extra activity of hidden neurons can be identified as noise.

Let us study a very simple example in the framework of section 3.5.2 where for any $\nu < 1/2$, the limit weights do not enable to correctly classify the objects.

We consider the set of object natures $\mathcal{O} = \{$◯, □, △, ◯, □, △, ◯, □, △$\}$. The set of features and input neurons is $I = \{$circle, square, triangle, blue, red, green$\}$. We assume that input neurons i spike independently of each other with probability p' if and only if the presented object has feature i, and we choose depth $L = 1$. Then any threshold $s_1 \in (0, p'^2)$ verifies Assumption 8 and in this case

$$\bar{J}_1 = \{j_{\{\text{blue,square}\}}, j_{\{\text{red,square}\}}, j_{\{\text{green,square}\}}, j_{\{\text{blue,circle}\}}, j_{\{\text{red,circle}\}}, j_{\{\text{green,circle}\}}, j_{\{\text{blue,triangle}\}}, j_{\{\text{red,triangle}\}}, j_{\{\text{green,triangle}\}}\}.$$

This set can be identified with the set of object natures $\mathcal{O}$, and Assumption 16 (binary correlations) is verified with $p = p'$. Therefore, any set of classes K verify Assumption 17 (class decomposition) since any class can be written as the union of its single elements. Let us choose $K = \{k_1, k_2\}$ with $k_1 = \{$□, □, ◯, △$\}$ and $k_2 = \{$□, ◯, ◯, △, △$\}$. This classes are such that class k_1 contains every item which is squared or blue except the blue square, and class k_2 contains every other possible object.

Proposition 20 *In this framework, for any $\nu \in [0, 1/2)$, the limit weights do not enable to correctly classify the* □.

Therefore, even with ν less but close to $1/2$, the limit network fails to correctly classify the objects, whereas for any $\nu \geq 1/2$ the results of section 3.5.2 hold and the limit network succeeds to accomplish the task. This proposition enlightens that even a small noise can stop the network from working properly in a very simple case. The choice of bias allows us to control this phenomenon.

4. Additional results in a more general framework

In this section, we examine the guarantees we can obtain about CHANI's behavior in a much more general setting than the CHANI EWA framework: the main assumption needed in this section is that the expert aggregation algorithm admits a regret bound. As each hidden and output neuron runs its own expert aggregation algorithm, it possesses its own regret. Through our choice of gains, we reinterpret these regrets, enabling us to gain meaningful insights into the network's learning.

4.1 Hidden layers

Assumption 21 (Hidden neurons regret bound) *There exists $C > 0$ such that for every $l \in [L]$, the expert aggregation algorithm f^l used to train the hidden layer of depth l is such that for any set of experts E and any deterministic sequence of gains $g^E_{[M]} := (g^e_m)_{e\in E, m\in[M]}$ taking value in $[a, b]$ with $a < b$,*

$$R_M \leq C(b-a)\sqrt{\ln(|E|)M}.$$

The interpretation of the regret of hidden neurons leads to the following result.

Proposition 22 (Hidden neurons regret bound) *Suppose Assumption 21 (hidden neurons regret bound) holds. Then a.s. for every $l \in [L]$ and $j = j_{S_1\cup S_2} \in J_l$ neuron of the l^{th} layer where S_1 and S_2 are feature subsets encoded in the previous layer,*

$$\max_{q\in\mathcal{P}_{j_{l-1}}} \left\langle (\psi^{j,l}_{m,t}(q) - \psi^{j,l}_{m,t}(w^j_m))X^{j_{S_1},l}_{m,t} X^{j_{S_2},l}_{m,t} \right\rangle_{\substack{t\in[T]\\ m\in[M]}} \leq C\sqrt{\frac{\ln(|\hat{J}_{l-1}|)}{M}}.$$

This regret bound can be interpreted as follows: since $p^{j,l}_{m,t}(w^j_m) = (\psi^{j,l}_{m,t}(w^j_m))_+$, on average, neuron $j = j_{S_1\cup S_2}$ will tend to spike as frequently as possible when neurons j_{S_1} and j_{S_2} are highly correlated, that is, when the product $X^{j_{S_1},l}_{m,t} X^{j_{S_2},l}_{m,t}$ is often equal to 1. Although this result has less implications than Theorem 9, it relies only on Assumption 21 (Hidden neurons regret bound). This makes it a significant step toward analyzing the role of hidden layers in CHANI within a very general setting.

4.2 Output layer

Assumption 23 (ξ-balanced) *There exists a constant $\xi > 0$ independent of N such that for every $k \in K$, $N_k/N \geq \xi$.*

This assumption means that during the training phase of the output layer, the proportion of presented objects of any class is at least ξ. This ensures that the network sees a significant number of objects of each class.

Proposition 24 *Suppose Assumptions 23 (ξ-balanced) and 11 (Ouptut neurons regret bound) hold. Then,*

$$\mathrm{Disc}_N(w^k_{[N]}, X^{\hat{J}_L,L+1}_{[N],[T]}) \geq \max_{q^K \in (\mathcal{P}_{\hat{J}_L})^{|K|}} \mathrm{Disc}_N(q^K, X^{\hat{J}_L,L+1}_{[N],[T]}) - \frac{C|K|}{\xi(|K|-1)}\sqrt{\frac{\ln(|\hat{J}_L|)}{N}} \quad a.s.$$

The proposition, established by combining local regret bounds of output neurons, ensures that the network discrepancy of CHANI exceeds the maximum achievable with constant weights for the output layer, minus an error term which tends to zero. Note that in this case the quantity $\mathrm{Disc}_N(q^K, X^{\hat{J}_L,L+1}_{[N],[T]})$, against which we compare CHANI's network discrepancy, depends on the activity of CHANI's final hidden layer. This is not the case in Corollary 12, which applies within the more restrictive CHANI EWA framework and relies on the present result as an initial step.

5. Numerical results

In this section, we present numerical results on two datasets: a simulated dataset consisting of colored shapes, for which the assumptions required by the theoretical results are satisfied, and the `digits` dataset from `scikit-learn`, for which not all assumptions hold, yet CHANI is nonetheless able to achieve strong performance.

5.1 Simulated dataset

We work in the framework of the example discussed in Section 3.6. We consider the set of object natures $\mathcal{O} = \{◯, □, △, ◯, □, △, ◯, □, △\}$ and the set of features $I = \{\text{circle}, \text{square}, \text{triangle}, \text{blue}, \text{red}, \text{green}\}$. The features are encoded in the input layer as follows: when presented with an object, the two input neurons coding for its shape and color spike independently with a fixed probability $p > 0$ and the other input neurons stay silent. We trained CHANI on two tasks.

• A first task where the classes are $k_1 = \{◯, ◯, ◯\}$ and $k_2 = \{□, △, □, △, □, △\}$. In this case, the classes can be described as combinations of single features (looking at the shape of the objects is sufficient to classify them): Assumption 17 (Class decomposition) is verified for the network without hidden layers, and we have

$$k_1 = \mathcal{O}^{\{\text{circle}\}} \quad \text{and} \quad k_2 = \mathcal{O}^{\{\text{square}\}} \cup \mathcal{O}^{\{\text{triangle}\}}.$$

• A second task where the classes are $k_1 = \{◯, △, □, ◯, □\}$ and $k_2 = \{□, ◯, △, △\}$. In this case, the classes cannot be described as combinations of single features, but as combinations of pairs of features: Assumption 17 (Class decomposition) is verified for the network with one hidden layer, and we have

$$k_1 = \mathcal{O}^{\{\text{circle,blue}\}} \cup \mathcal{O}^{\{\text{triangle,blue}\}} \cup \mathcal{O}^{\{\text{square,red}\}} \cup \mathcal{O}^{\{\text{circle,green}\}} \cup \mathcal{O}^{\{\text{square,green}\}}$$

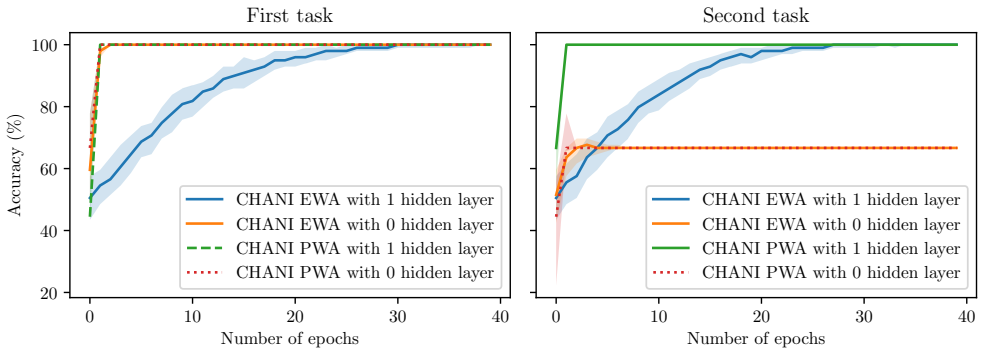

Figure 3: Accuracy with confidence interval of level 0.9 *on simulated dataset for CHANI with EWA and CHANI with PWA, with zero and one hidden layer. The parameters are* $T = 2000$, $M = 40$, $N = 360$, $p = 0.5$ *and each configuration was run* 100 *times. For CHANI with EWA and zero hidden layer:* $\eta^1 = 0.05$. *For CHANI with EWA and one hidden layer:* $\eta^1 = 3$ *and* $\eta^2 = 0.05$. *For CHANI with PWA:* $b = 2$. *For CHANI with EWA and PWA with one hidden layer:* $s_1 = 0.1$. *For each configuration, the training data are organized into epochs, where one epoch corresponds to a sequence of the* 9 *object natures presented in random order. After each epoch of the output layer training, the network's accuracy is evaluated on a test set consisting of* 99 *objects.*

and

$$k_2 = \mathcal{O}^{\{\text{square,blue}\}} \cup \mathcal{O}^{\{\text{circle,red}\}} \cup \mathcal{O}^{\{\text{triangle,red}\}} \cup \mathcal{O}^{\{\text{triangle,green}\}}.$$

On Figure 3 is represented the accuracy with confidence interval of level 0.9 accross training for several configurations of the network: CHANI with either EWA or PWA as expert aggregation algorithm, and with zero or one hidden layer.

For the first task, which is theoretically achievable by CHANI EWA with zero or one hidden layer according to Theorem 18, we can see that every method achieves 100% accuracy. However, CHANI EWA or PWA with zero hidden layer and CHANI PWA with one hidden layer learn faster and with less variance than CHANI EWA with one hidden layer.

For the second task, which is theoretically achievable by CHANI EWA with one hidden layer and not by CHANI EWA with zero hidden layer according to Theorem 18, we can see that CHANI EWA or PWA with zero hidden layer does not exceed 67% accuracy, whereas CHANI with EWA or PWA with one hidden layer reaches 100%. However, CHANI PWA is still faster to learn and with less variance in this case.

5.2 Handwritten digits dataset

We trained CHANI on the `digits` dataset of `scikit-learn`. This dataset provides 1797 images of 64 pixels of handwritten digits which were randomly separated

in a training set (80% of the images) and a testing set (20% of the images). For computational reasons, we decided to use this dataset rather than one with better resolution. The numerical results for CHANI with no hidden layer (which corresponds to the network of our previous work HAN) are visible in Table 1. Although CHANI with EWA performs much better than CHANI with PWA, its accuracy is only around 50%.

Expert Aggregation algorithm	Accuracy	Confidence interval of level 0.9
EWA	53.0	$[50.8, 55.6]$
PWA	14.8	$[14.7, 15.0]$

Table 1: Numerical results with no hidden layer. The parameters are $T = 2000$*,* $N = 1437$*,* $\eta^1 = 0.0005$*,* $b = 2$*. We made* 100 *realizations.*

The results for CHANI with one hidden layer are visible in Figure 4. The blue curve (resp. the green curve) represents the percentage of correct classifications of the testing set of CHANI with the expert aggregation algorithm EWA (resp. PWA) in function of the number of selected hidden neurons. Before selection, there was 2016 neurons. We can see that no matter the number of selected neurons, CHANI with PWA performs badly and does not exceed a 40% correct classifications. Besides, its performance is unstable. On the other hand, the performance of CHANI with EWA increases steadily with the number of selected neurons, to station at around 83.5%, and exceeds 80% from 80 selected neurons. This suggests that for a more complex task than that of the previous section, the expert aggregation EWA performs better.

The numerical results with two hidden layers and EWA can be seen in Table 2. Here, we fixed the number of selected neurons of the first hidden layer and varied the number of selected neurons of the second one. We can see that in all cases, the results, comparable to CHANI with no hidden layer, are far less good than with one hidden layer. There are several possible explanations. A good representation of classes could be combination of correlation between two features and not three (*i.e.,* a digit could be well represented by combinations of pixel tuples and not triplets). Numerical results could also be affected by the phenomenon of decreasing spiking probability from one layer to the next: there may not be enough spikes in the system in order to have a second hidden layer with meaningful cumulated gains. Another reason could be that this dataset does not respect all the assumptions that we made in order to have theoretical guarantees.

Addition of connections. In order to improve CHANI performance, we added connections between input and output neurons in the case $L = 1$. Indeed, a digit could be well represented by the combination of single pixels and pixel tuples.

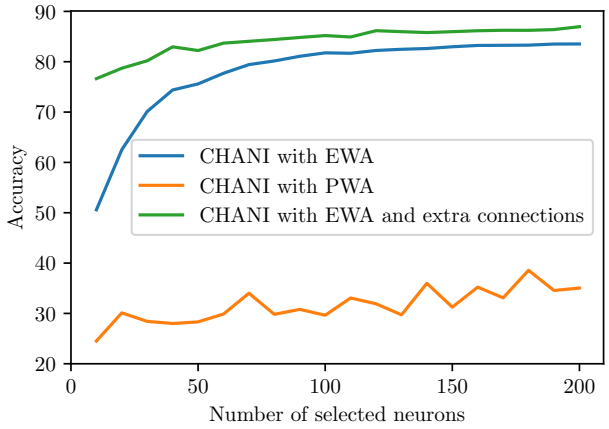

Figure 4: Numerical results on digits dataset with one hidden layer for CHANI with EWA, CHANI with PWA and CHANI with EWA and extra connections. The parameters $T = 2000$, $M = 40$, $N = 1357$ *and* 20 *realizations were made for each number of selected neurons. For CHANI with EWA:* $\eta^1 = 3$, $\eta^2 = 0.002$. *For CHANI with PWA:* $b = 2$ *for each layer. For CHANI with EWA and extra connections:* $\eta^1 = 3$, $\eta^2 = 0.007$, $\alpha = 0.7$ *and* $\beta = 0.25$.

Number of selected neurons on the second hidden layer	Accuracy
50	44.6
100	49.7
150	51.0

Table 2: Numerical results with two hidden layers for CHANI with EWA. The parameters are $T = 2000$, $M = 40$, $N = 1277$, $\eta^3 = 0.0005$. *We made* 10 *realizations, and for each we selected* 100 *of hidden neurons in the first hidden layer.*

To do this, we extend the gains of hidden neurons w.r.t. output neurons defined by (7) to input neurons as well, and we add a multiplicative parameter $\alpha > 0$ in order to find balance between hidden and input neurons gains. Then the gain of an input neuron i w.r.t. output neuron k when presented with object o_m^{L+1} is

$$g_m^{j\to k} := \begin{cases} \alpha \langle X_{m,t}^{i,L+1} \rangle_{t\in[T]} \times \frac{N}{N^k} & \text{if } o_m^{L+1} \in k \\ -\alpha \langle X_{m,t}^{i,L+1} \rangle_{t\in[T]} \times \frac{N}{N^{k'}} \times \frac{1}{|K|-1} & \text{if } o_m^{L+1} \in k' \neq k \end{cases}.$$

Indeed, as the neurons spiking probabilities decrease from one layer to the next, the empirical spiking probabilities of neurons from distinct layers may have very different order of magnitude. For the same reason, we introduce a parameter $\beta \in [0, 1]$ as well in the formula of the conditional spiking probability of an output

neuron k when presented with object o_m^{L+1}:

$$p_{m,t}^k(w_m^k) = w_m^{\hat{J}_L \to k} \cdot X_{m,t-1}^{\hat{J}_1, L+1} + \beta w_m^{I \to k} \cdot X_{m,t-1}^{I, L+1}$$

where $w_m^k := (w_m^{l \to k})_{l \in I \cup \hat{J}_L}$, $w_m^{\hat{J}_L \to k} := (w_m^{j \to k})_{j \in \hat{J}_L}$ and $w_m^{I \to k} := (w_m^{i \to k})_{i \in I}$. This parameter β allows to reduce the impact of input neurons, which spike more frequently than hidden neurons, on the behavior of output neurons.

The numerical results with this configuration and the expert aggregation EWA correspond to the green curve of Figure 4. We can see that CHANI performs better with these new connections: with only 10 selected neurons, its accuracy on the testing set is already at 76.6% and it grows reach 87% for 200 selected neurons, and exceeds 84% from 70 selected neurons.

Comparison with STDP: The same dataset has been used to train spiking neural networks with STDP by Rybka et al. (2024); Sboev et al. (2022). Their results vary from 83% to 95% of accuracy depending on the network settings: in some of them they are comparable to CHANI's performance. Note that these models are more complex and do not have any theoretical guarantees.

To conclude, CHANI with EWA succeeds the learning task with reasonable results. This is a huge improvement in comparison to our numerical results of Jaffard et al. (2024), where we trained HAN solely on the dataset of section 5.1. By adding layers and taking neuronal synchronization into account, CHANI can now learn more realistic tasks.

Conclusion

In this paper, we introduced our cognitive network, CHANI (Correlation-based Hawkes Aggregation of Neurons with bio-Inspiration), which provably learns to classify objects, and we presented theoretical insights into CHANI's learning capabilities with any number of hidden layers. First, we interpreted regret bounds resulting from the use of expert aggregation algorithms as learning rule from a learning perspective. Subsequently, in the specific framework CHANI EWA, we demonstrated that at the limit hidden neurons encode feature correlations, and we proved that our neuron selection method prioritizes those encoding the most significant correlations. Furthermore, through an analysis of the asymptotic behavior of output neurons, we demonstrated CHANI's ability to learn the correct classification of objects on average and even asymptotically when classes are defined by feature correlations. Lastly, we calculated the VC-dimension of our network. To our knowledge, this study is the first one to establish that local learning rules enable a biologically inspired network to learn to recognize complex concepts by forming neuronal assemblies, inducing global learning. However, several challenging

research avenues remain unexplored. In order to make our model more realistic and computationally efficient, one interesting direction is investigating CHANI's behavior when objects are not presented for a fixed duration, but only until one output neuron has spiked significantly more than the others. This would correspond to a reaction time and would have a cognitive interpretation. Another innovative line of research is extending these findings by establishing regret bounds for other local rules, such as STDP (Spike-Timing-Dependent Plasticity).

Acknowledgments

This research was supported by the French government, through CNRS (eXplAIn team, MITI interdisciplinary program and Algernon team (CNRS Interinstitute PRIME Label)), the UCA^{Jedi} and 3iA Côte d'Azur Investissements d'Avenir managed by the National Research Agency (ANR-15 IDEX-01 and ANR-19-P3IA-0002), directly by the ANR project ChaMaNe (ANR-19-CE40-0024-02) and GraVa (ANR-18-CE40-0005), and finally by the interdisciplinary Institute for Modeling in Neuroscience and Cognition (NeuroMod).

All the numerical results can be reproduced thanks to the code available at `https://github.com/SophieJaffard/CHANI`. In Appendix A, we summarize all the notations and in Appendix B and C we provide the proofs of our results. We do not provide the proofs in the order in which the results are enunciated in the article, but from the most general results to the most specific ones.

Appendix A. Notations

Table 3: Table of notations

Notation	Description
$\mathcal{O}$	set of objects
o	nature of an object
M	number of objects presented to the network to train a hidden layer
N	number of objects presented to the network to train the output layer
m	index of an object
T	number of time steps during which one object is presented
I	set of features and input neurons
i	index of an input neuron or a feature
K	set of classes and output neurons
k	index of an output neuron or a class
L	number of hidden layers
l	depth of a hidden layer
I_l	set of every subsets of I having cardinal 2^l
J_l	set of hidden neurons of depth l before selection
$\hat{J}_l$	set of selected hidden neurons of depth l
$\bar{J}_l$	set of hidden neurons of interest of depth l
$j = j_S$	index of a hidden neuron coding for a set $S \subset I$
o_m^l	nature of the m^{th} object of the training phase of layer l
$\mathcal{P}_E$	set of probability distributions over the set E

Appendix B. Proofs of section 4

B.1 Proof of Proposition 22

Let $l \in [L]$, $j = j_S \in J_l$, where $S = S_1 \cup S_2 \subset I$ with S_1 and S_1 encoded in the previous layer. Then

$$\sum_{m=1}^{M} w_m^j \cdot g_m^j = \sum_{m=1}^{M} w_m^j \cdot (\hat{\rho}_m^l(\{j_{S_1}, j_{S_2}, j'\})_{j' \in \hat{J}_{l-1}}$$

$$= \sum_{m=1}^{M} w_m^j \cdot (\langle X_{m,t}^{j_{S_1},l} X_{m,t}^{j_{S_2},l} X_{m,t}^{j',l} \rangle_{t\in[T]})_{j'\in\hat{J}_{l-1}}$$
$$= \sum_{m=1}^{M} \langle X_{m,t}^{j_{S_1},l} X_{m,t}^{j_{S_2},l} (w_m^j \cdot X_{m,t}^{\hat{J}_{l-1},l}) \rangle_{t\in[T]}$$

and

$$\sum_{m=1}^{M} w_m^j \cdot g_m^j + \sum_{m=1}^{M} \langle -\nu X_{m,t}^{j_{S_1},l} X_{m,t}^{j_{S_2},l} \rangle_{t\in[T]} = \sum_{m=1}^{M} \langle X_{m,t}^{j_{S_1},l} X_{m,t}^{j_{S_2},l} \psi_{m,t}^{j,l}(w_m^j) \rangle_{t\in[T]}.$$

Similarly, for $q \in \mathcal{P}_{\hat{J}_{l-1}}$,

$$\sum_{m=1}^{M} q \cdot g_m^j + \sum_{m=1}^{M} \langle -\nu X_{m,t}^{j_{S_1},l} X_{m,t}^{j_{S_2},l} \rangle_{t\in[T]} = \sum_{m=1}^{M} \langle X_{m,t}^{j_{S_1},l} X_{m,t}^{j_{S_2},l} \psi_{m,t}^{j,l}(q) \rangle_{t\in[T]}.$$

Hence the regret of neuron j is $R_M^j = \max_{q\in\mathcal{P}_{\hat{J}_{l-1}}} \sum_{m=1}^{M} \left\langle X_{m,t}^{j_{S_1}} X_{m,t}^{j_{S_2}} (\psi_{m,t}^{j,l}(q) - \psi_{m,t}^{j,l}(w_m^j)) \right\rangle_{t\in[T]}$. The gains take value in $[0,1]$ so we get the bound by applying the regret bound of the expert aggregation given by Assumption 21.

B.2 Proof of Proposition 24

Let $k \in K$, $j \in \hat{J}_L$. Then

$$\frac{1}{N}\sum_{m=1}^{N} w_m^k \cdot g_m^k = \frac{1}{N^k} \sum_{m, o_m^{L+1}\in k} w_m^k \cdot \langle X_{m,t}^{j,L+1} \rangle_{t\in[T]}$$
$$- \left\langle \frac{1}{N^{k'}} \sum_{m, o_m^{L+1}\in k'} w_m^k \cdot \langle X_{m,t}^{j,L+1} \rangle_{t\in[T]} \right\rangle_{k' \ / \ k'\neq k}$$
$$= \mathrm{Disc}_N^k(w_{[N]}^k, X_{[N],[T]}^{\hat{J}_L,L+1})$$

and similarly for $q^k \in \mathcal{P}_{\hat{J}_{L-1}}$, we have $\frac{1}{N}\sum_{m=1}^{N} q^k \cdot g_m^k = \mathrm{Disc}_N^k(q^k, X_{[N],[T]}^{\hat{J}_L,L+1})$. So the regret of neuron k divided by N is

$$\frac{R_N^k}{N} = \max_{q^k\in\mathcal{P}_{\hat{J}_{L-1}}} \mathrm{Disc}_N^k(q^k, X_{[N],[T]}^{\hat{J}_L,L+1}) - \mathrm{Disc}_N^k(w_{[N]}^k, X_{[N],[T]}^{\hat{J}_L,L+1}) \leq \frac{C|K|}{\xi(|K|-1)} \sqrt{\frac{\ln(|\hat{J}_L|)}{N}} \tag{12}$$

according to Assumption 11, because the gains take value in $[-\frac{1}{\xi(|K|-1)}, \frac{1}{\xi}]$. Then

$$\mathrm{Disc}_N(w_{[N]}^K, X_{[N],[T]}^{\hat{J}_L,L+1}) = \left\langle \hat{p}_m^k(w_m^k) - \hat{p}_m^{k'}(w_m^{k'}) \right\rangle_{\substack{k\in K \\ k'\neq k \\ m, o_m^{L+1}\in k}}$$

$$= \Big\langle \hat{p}_m^k(w_m^k) \Big\rangle_{\substack{k\in K \\ m, o_m^{L+1}\in k}} - \Big\langle \hat{p}_m^{k'}(w_m^{k'}) \Big\rangle_{\substack{k\in K \\ k'\neq k \\ m, o_m^{L+1}\in k}}$$

Let us exchange the name of the indexes k and k' in the second term.

$$\mathrm{Disc}_N(w_{[N]}^K, X_{[N],[T]}^{\hat{J}_L,L+1}) = \Big\langle \hat{p}_m^k(w_m^k) \Big\rangle_{\substack{k\in K \\ m, o_m^{L+1}\in k}} - \Big\langle \hat{p}_m^k(w_m^k) \Big\rangle_{\substack{k\in K \\ k'\neq k \\ m, o_m^{L+1}\in k'}}$$

$$= \Big\langle \langle \hat{p}_m^k(w_m^k)\rangle_{m, o_m^{L+1}\in k} - \langle \hat{p}_m^k(w_m^k)\rangle_{\substack{k'\neq k \\ m, o_m^{L+1}\in k'}} \Big\rangle_{k\in K} = \langle \mathrm{Disc}_N^k(w_{[N]}^k, X_{[N],[T]}^{\hat{J}_L,L+1})\rangle_{k\in K}.$$

and similarly $\mathrm{Disc}_N(q^K, X_{[N],[T]}^{\hat{J}_L,L+1}) = \langle \mathrm{Disc}_N^k(q^k, X_{[N],[T]}^{\hat{J}_L,L+1})\rangle_{k\in K}$. Therefore, according to (12),

$$\mathrm{Disc}_N(w_{[N]}^K, X_{[N],[T]}^{\hat{J}_L,L+1}) \geq \max_{q^K\in \mathcal{P}_{\hat{J}_L}^{|K|}} \mathrm{Disc}_N(q^K, X_{[N],[T]}^{\hat{J}_L,L+1}) - \frac{C|K|}{\xi(|K|-1)}\sqrt{\frac{\ln(|\hat{J}_L|)}{N}}.$$

Appendix C. Proofs of section 3

C.1 Preliminary propositions

Proposition 25 *Let $(\Omega, \mathcal{F}, \mathbb{P})$ be a probability space, let $\mathcal{G} \subset \mathcal{F}$ be a σ-algebra, and let E, F be $\mathcal{G}$-measurable random finite sets.*

Let $A, B \in \mathbb{N}^*$, and let $(Z_{a,b}^{e\to f})_{e\in E, f\in F, 1\leq a\leq A, 1\leq b\leq B}$ be random variables bounded by 1 such that for every $f \in F$, $e \in E$ the variables $(Z_{a,b}^{e\to f})_{1\leq a\leq A, 1\leq b\leq B}$ are independent knowing the σ-algebra $\mathcal{G}$. Let $\alpha > 0$. Then with probability $1-\alpha$, for all $e \in E, f \in F$

$$\left| \sum_{a=1}^{A} \langle Z_{a,b}^{e\to f}\rangle_{b\in[B]} - \mathbb{E}\Big[\sum_{a=1}^{A} \langle Z_{a,b}^{e\to f}\rangle_{b\in[B]} \mid \mathcal{G}\Big] \right| < \sqrt{\frac{A}{2B}\ln\Big(\frac{2|E||F|}{\alpha}\Big)}.$$

Proof Let $\alpha > 0$, $e \in E$, $f \in F$. The variables $(Z_{a,b}^{e\to f})_{1\leq a\leq A, 1\leq b\leq B}$ are independent knowing the σ-algebra $\mathcal{G}$ and bounded by 1 so according to Hoeffding's inequality, for $\beta > 0$ and $\mathcal{G}$-measurable,

$$\mathbb{P}\left(\left| \sum_{a=1}^{A} \langle X_{a,b}^{e\to f}\rangle_{b\in[B]} - \mathbb{E}\Big[\sum_{a=1}^{A} \langle X_{a,b}^{e\to f}\rangle_{b=1,\dots,B} \mid \mathcal{G}\Big] \right| \geq \beta \mid \mathcal{G} \right) \leq 2e^{-2\beta^2 \frac{B}{A}}.$$

Let

$$D^c := \Big\{ \exists e \in E, f \in F, \left| \sum_{a=1}^{A} \langle X_{a,b}^{e\to f}\rangle_{b=1,\dots,B} - \mathbb{E}\Big[\sum_{a=1}^{A} \langle X_{a,b}^{e\to f}\rangle_{b\in[B]} \mid \mathcal{G}\Big] \right| \geq \beta \Big\}.$$

Then $\mathbb{P}(D^c \mid \mathcal{G}) \leq |E||F| 2e^{-2\beta^2 \frac{B}{A}}$. Let us choose $\beta = \sqrt{\frac{A}{2B} \ln\left(\frac{2|E||F|}{\alpha}\right)}$. Then $\mathbb{P}(D^c \mid \mathcal{G}) \leq \alpha$. By integrating we get $\mathbb{P}(D^c) \leq \alpha$.

■

Proposition 26 *Let $d \in \mathbb{N}$, $A^1, \dots, A^d \in \mathbb{R}$. Let E be the subset of $[d]$ defined by $E := \arg\max_i A^i$, and let $w_i(M) := \frac{\exp(\sqrt{M}A^i)}{\sum_{l=1}^d \exp(\sqrt{M}A^l)}$. Let $A = \max_i A^i$.*

- *If $E = [d]$ then for every $i \in [d]$, $w_i(M) = \frac{1}{d}$.*
- *If $E \subsetneq [d]$, let $\Delta := \max_i A^i - \max_{i \notin E} A^i$. Then, for all i,*

$$\left| w_i(M) - \frac{1}{|E|} \mathbb{1}_{i \in E} \right| \leq \frac{1}{|E|} \max\left(1, \frac{d - |E|}{|E|}\right) \exp(-\sqrt{M}\Delta).$$

Proof Let $A_{\max} := \max_i A^i$.
Case $E = [d]$. Then for all $i \in [d]$

$$w_i(M) = \frac{\exp(\sqrt{M}A_{\max})}{\sum_{l=1}^d \exp(\sqrt{M}A_{\max})} = \frac{1}{d}.$$

Case $E \subsetneq [d]$. Let $A_{\max \text{ bis}} := \max_{i \notin E} A^i$. Then

$$w_i(M) \leq \frac{\exp(\sqrt{M}A^i)}{|E| \exp(\sqrt{M}A_{\max})} = \frac{1}{|E|} \exp(-\sqrt{M}(A_{\max} - A^i)) \tag{13}$$

Let $i \in E$.

$$\begin{aligned} w_i(M) &\geq \frac{\exp(\sqrt{M}A_{\max})}{|E| \exp(\sqrt{M}A_{\max}) + (d - |E|) \exp(\sqrt{M}A_{\max \text{ bis}})} \\ &= \frac{1}{|E|} \frac{1}{1 + \frac{d-|E|}{|E|} \exp(-\sqrt{M}(A_{\max} - A_{\max \text{ bis}}))} \\ &\geq \frac{1}{|E|} \Big(1 - \frac{d - |E|}{|E|} \exp(-\sqrt{M}(A_{\max} - A_{\max \text{ bis}}))\Big) \\ &= \frac{1}{|E|} - \frac{d - |E|}{|E|^2} \exp(-\sqrt{M}(A_{\max} - A_{\max \text{ bis}})) \end{aligned}$$

And thanks to (13), $w_i(M) \leq \frac{1}{|E|}$. Hence

$$\frac{1}{|E|} - \frac{d - |E|}{|E|^2} \exp(-\sqrt{M}(A_{\max} - A_{\max \text{ bis}})) \leq w_i(M) \leq \frac{1}{|E|}.$$

Let $i \in [d] \setminus E$. According to (13)

$$0 \leq w_i(M) \leq \frac{1}{|E|} \exp(-\sqrt{M}(A_{\max} - A_{\max \text{ bis}})).$$

■

As proved in (Gao and Pavel, 2018, Proposition 4), we have

Proposition 27 *Let $d \in \mathbb{N}$, $\eta \in \mathbb{R}$. Then the softmax function f with temperature η defined on $\mathbb{R}^d$ by $x \mapsto \text{softmax}(\eta x)$ is η-Lipschitz.*

C.2 Definition of the network activity and coupled network activity

Let us define more precisely the variables $X_{m,t}^{j,l}$ and $X_{o,t}^{j,l}$. For $l \in [L]$, let $\mathcal{F}_l$ be the $\sigma-$algebra generated by every event that happened until the end of the selection phase of layer l.

• For $i \in I, m \in [M]$, $l \in [L]$, the variables $(X_{m,t}^{i,l})_{1 \leq t \leq T}$ are i.i.d and follow a Bernoulli distribution with parameter p_o^i where $o_m^l = o$. Similarly, for $i \in I, o \in \mathcal{O}$, $l \in [L]$, the variables $(X_{o,t}^{i,l})_{1 \leq t \leq T}$ are i.i.d and follow a Bernoulli distribution with parameter p_o^i.

• For $l \in [L]$, $j \in \hat{J}_1 \cup \cdots \cup \hat{J}_{l-1} \cup J_l$, we define conditionally to the past $\mathcal{F}_{l-1}$ the variables $(U_{m,t}^{j,l})_{1 \leq m \leq M, 1 \leq t \leq T}$ and $(U_{o,t}^{j,l})_{o \in \mathcal{O}, 1 \leq t \leq T}$ which are i.i.d and follow a uniform distribution on $[0,1]$. Then, if $j \notin J_l$ we define its activity by $X_{m,t}^{j,l} := \mathbb{1}_{p_{m,t}^{j,l}(w_{M+1}^j) \geq U_{m,t}^{j,l}}$, if $j \in J_l$ then $X_{m,t}^{j,l} := \mathbb{1}_{p_{m,t}^{j,l}(w_m^j) \geq U_{m,t}^{j,l}}$, and in any case $X_{o,t}^{j,l} := \mathbb{1}_{p_{o,t}^{j,l}(w_{M+1}^j) \geq U_{o,t}^{j,l}}$.

Let $l \in [L]$, $l' \leq l$ and $w_{\text{bis}} := (w_{\text{bis}}^j)_{j \in \hat{J}_1 \cup \cdots \cup \hat{J}_{l-1} \cup J_l}$ an $\mathcal{F}_{l-1}$-measurable weight family. Then we define the coupled network activity with weights w_{bis} as the variables $Z_{m,t}^{j,l} := \mathbb{1}_{p_{m,t}^{j,l}(w_{\text{bis}}^j) \geq U_{m,t}^{j,l}}$ and $Z_{o,t}^{j,l} := \mathbb{1}_{p_{o,t}^{j,l}(w_{\text{bis}}^j) \geq U_{o,t}^{j,l}}$.

C.3 Proof of Theorem 9

We recall that for $l \in [L]$, $j = j_{S_1 \cup S_2} \in \bar{J}_l$, we denote $\bar{w}^j = \mathbb{1}_{j \in \{j_{S_1}, j_{S_2}\}}$. Let $\alpha > 0$. For $l \geq 1$ and $j = j_S$ where $S \subset I$, let

$$E_{\text{EWA}}^{j,l}(M, |\bar{J}_{l-1}|) := \frac{1}{2} \max(1, \frac{|\bar{J}_{l-1}| - 2}{2}) \exp\Big(- \rho(S) 2^{2-2^l} \sqrt{2M \ln(|\bar{J}_{l-1}|)}\Big).$$

For $A, B \geq 1$, let

$$E_{\text{approx}}(T, A, B, \alpha) := 2\sqrt{\frac{A \ln(A)}{T} \ln\Big(\frac{2AB}{\alpha}\Big)}.$$

For $l \in [L]$ let

$$E_w(l) := \max_{j \in \bar{J}_l} \sqrt{|\bar{J}_{l-1}|} \|w_{M+1}^j - \bar{w}^j\|_2$$

and

$$E_{\text{prec}}(M, |\bar{J}_l|, l) := 6\sqrt{2\ln(|\bar{J}_l|)M}\sum_{l'=1}^{l} E_w(l').$$

Moreover, we denote $\bar{J}_l^* := \{j_{S_1\cup S_2} \text{ such that } j_{S_1}, j_{S_2} \in \bar{J}_l \text{ and } S_1 \cap S_2 = \emptyset\}$. Let us prove by induction the following proposition for every $l \in [L]$.

Proposition 28 *There exist constants C_1^l and C_2^l and events $(D^{l'})_{l'\le l}$ and $(D_{sel}^{l'})_{l'\le l}$, each of probability more than $1-\alpha$, such that if $T/M^{l-1} \ge C_1^l$ and $M \ge C_2^l$ then on $D^1 \cap D_{sel}^1 \cap \cdots \cap D^{l-1} \cap D_{sel}^l$, for every $l' \le l$, $\hat{J}_{l'} = \bar{J}_{l'}$ and for all $j \in \bar{J}_{l'}$, $j' \in \bar{J}_{l'-1}$,*

$$|w_{M+1}^{j'\to j} - \bar{w}^{j'\to j}| \le E_{approx}(T, |\bar{J}_{l'-1}|, |\bar{J}_{l'-1}^*|, \alpha) + E_{EWA}^{j,l}(M, |\bar{J}_{l'-1}|) + E_{prec}(M, |\bar{J}_{l'-1}|, l'-1)$$

Case l=1: For $i \in I$, $j = j_{\{i_1,i_2\}} \in J_1$, we remind that the cumulated gain of neuron i w.r.t. neuron j is $G_M^{i\to j} = \sum_{m=1}^M \langle X_{m,t}^{i,1} X_{m,t}^{i_1,1} X_{m,t}^{i_2,1}\rangle_{t\in[T]}$. Let

$$\bar{G}_M^{i\to j} := \sum_{m=1}^{M} \mathbb{P}(X_{m,t}^{i,1} = 1, X_{m,t}^{i_1,1} = 1, X_{m,t}^{i_2,1} = 1).$$

The variables $(X_{m,t}^{i,1} X_{m,t}^{i_1,1} X_{m,t}^{i_2,1})_{1\le m\le M, 1\le t\le T}$ are independent so according to Proposition 25 there exists an event D^1 of probability more than $1-\alpha$ on which for all $j \in J_1$, $i \in I$

$$|G_M^{i\to j} - \bar{G}_M^{i\to j}| \le \sqrt{\frac{M}{2T}\ln\Big(\frac{2|J_1||I|}{\alpha}\Big)}.$$

Let $\bar{w}_{M+1}^{i\to j} := \frac{\exp(\eta_1 \bar{G}_M^{i\to j})}{\sum_{i'\in I}\exp(\eta_1 \bar{G}_M^{i'\to j})}$. Then according to (27),

$$\|w_{M+1}^j - \bar{w}_{M+1}^j\|_2 \le \eta_1 \sqrt{\sum_{i'\in I} |G_M^{i'\to j} - \bar{G}_M^{i'\to j}|}.$$

Hence on D^1, with $\eta_1 = \sqrt{\frac{8\ln(|I|)}{M}}$ we get

$$\|w_{M+1}^j - \bar{w}_{M+1}^j\|_2 \le 2\sqrt{\frac{|I|\ln(|I|)}{T}\ln\Big(\frac{2|I||J_1|}{\alpha}\Big)} = E_{\text{approx}}(T, |I|, |J_1|, \alpha). \quad (14)$$

Besides, for $j = j_{\{i_1,i_2\}} \in J_1$ and $i \in I$, $\bar{G}_M^{i\to j} = \sum_{m=1}^M \rho_{o_m^1}(\{i, i_1, i_2\})$. According to Assumption 5, each nature of object is presented the same amount of times so

$$\bar{G}_M^{i\to j} = \frac{M}{|\mathcal{O}|}\sum_{o\in\mathcal{O}} \rho_o(\{i, i_1, i_2\}) = M\rho(\{i, i_1, i_2\}).$$

and $\eta_1 \bar{G}_M^{i\to j} = \sqrt{8\ln(|I|)M}\rho(\{i,i_1,i_2\})$. We can distinguish between two cases.
1. If $j = j_{\{i_1,i_2\}}$ is such that $\rho(\{i_1,i_2\}) = 0$ then for all $i \in I, \rho(\{i,i_1,i_2\}) = 0$. Hence according to Proposition 26, we have

$$\bar{w}_{M+1}^{i\to j} = \bar{w}^{i\to j} \tag{15}$$

where $\bar{w}^{i\to j} := \frac{1}{|I|}$. This definition of $\bar{w}^{i\to j}$ does not conflict with the one given at beginning of the proof because according to the definition of $\bar{J}_1$ given in section 3.1, $j \notin \bar{J}_1$.
2. If $j = j_{\{i_1,i_2\}}$ is such that $\rho(\{i_1,i_2\}) > 0$ then according to Assumption 7, $\arg\max_{i\in I} \rho(\{i,i_1,i_2\}) = \{i_1,i_2\}$. Hence according to Proposition 26, we have

$$|\bar{w}_{M+1}^{i\to j} - \bar{w}^{i\to j}| \leq E_{\text{EWA}}^{j,1}(M,|I|). \tag{16}$$

Hence at the end of the learning phase of layer 1, by combining (14) and (15) we have for all j_S such that $\rho(S) = 0$:

$$\|w_{M+1}^j - \bar{w}^j\|_2 \leq E_{\text{approx}}(T,|I|,|J_1|,\alpha) \tag{17}$$

and by combining (14) and (16) we have for all j_S such that $\rho(S) > 0$:

$$\|w_{M+1}^j - \bar{w}^j\|_2 \leq E_{\text{approx}}(T,|I|,|J_1|,\alpha) + \sqrt{|I|}E_{\text{EWA}}^{j,1}(M,|I|). \tag{18}$$

Now let us study the selection phase. Let $\mathcal{F}_1^{\text{learn}} := \mathcal{F}_{M,T}^1$ be the σ-algebra generated by every event that happened until the end of the learning phase of layer 1. Let $j \in J_1$. The variables $(X_{o,t}^{j,1})_{o\in\mathcal{O},1\leq t\leq T}$ are independent knowing the σ-algebra $\mathcal{F}_1^{\text{learn}}$ since the weights w_{M+1}^j are frozen and are $\mathcal{F}_1^{\text{learn}}$-measurable. Hence according to Proposition 25, there exists a set D_{sel}^1 of probability more than $1-\alpha$ on which for every $j \in J_1$, $o \in \mathcal{O}$,

$$\left|\langle X_{o,t}^{j,1}\rangle_{t\in[T]} - \mathbb{E}[X_{o,1}^{j,1} \mid \mathcal{F}_1^{\text{learn}}]\right| \leq \sqrt{\frac{1}{2T}\ln\Big(\frac{2|J||\mathcal{O}|}{\alpha}\Big)}$$

since $\mathbb{E}[X_{o,t}^{j,1} \mid \mathcal{F}_1^{\text{learn}}]$ does not depend on t.

Let $(Z_{o,t}^{j,1})_{j\in J_1,o\in\mathcal{O},1\leq t\leq T}$ be the coupled network activity with weights $(\bar{w}^j)_{j\in J_1}$ as defined in section C.2. Then for $j \in J_1$, $o \in \mathcal{O}$,

$$\mathbb{E}[|X_{o,t}^{j,1} - Z_{o,t}^{j,1}| \mid \mathcal{F}_1^{\text{learn}}] \leq |p_{o,t}^{j,1}(w_{M+1}^j) - p_{o,t}^{j,1}(\bar{w}^j)| \leq |(w_{M+1}^j - \bar{w}^j)\cdot X_{o,t}^{I,1}|.$$

By Cauchy-Schwartz inequality and since the variables $X_{o,t}^{i,1}$ are bounded by 1,

$$\mathbb{E}[|X_{o,t}^{j,1} - Z_{o,t}^{j,1}| \mid \mathcal{F}_1^{\text{learn}}] \leq \|w_{M+1}^j - \bar{w}^j\|_1 \leq \sqrt{|I|}\|w_{M+1}^j - \bar{w}^j\|_2 = E_w(1).$$

Therefore, on D^1_{sel}, for all $j \in J_1$ and $o \in \mathcal{O}$

$$\left|\langle X^{j,1}_{o,t}\rangle_{t\in[T]} - \mathbb{E}[Z^j_{o,1} \mid \mathcal{F}^{\text{learn}}_1]\right| \leq \sqrt{\frac{1}{2T}\ln\Big(\frac{2|J||\mathcal{O}|}{\alpha}\Big)} + E_w(1).$$

Let $j = j_{\{i_1,i_2\}} \in J_1$. Let us compute $\mathbb{E}[Z^j_{o,1} \mid \mathcal{F}^{\text{learn}}_1]$. If $\rho(\{i_1,i_2\}) > 0$ then

$$\mathbb{E}[Z^j_{o,1} \mid \mathcal{F}^{\text{learn}}_1] = \mathbb{E}\Big[\Big(-\frac{1}{2} + \frac{1}{2}(X^{i_1,1}_{o,1} + X^{i_2,1}_{o,1})\Big)_+\Big] = \frac{1}{2}\rho_o(\{i_1,i_2\}).$$

If $\rho(\{i_1,i_2\}) = 0$ then $\mathbb{E}[Z^j_{o,1} \mid \mathcal{F}^{\text{learn}}_1] = \mathbb{E}\Big[\Big(-\frac{1}{2} + \frac{1}{|I|}\sum_{i\in I} X^{i,1}_{o,1}\Big)_+\Big] = 0 = \frac{1}{2}\rho_o(\{i_1,i_2\})$ according to Assumption 8 because less than half the neurons of I are active when presented with o and because $\rho_o(\{i_1,i_2\}) = 0$. Therefore on D^1_{sel}

$$\left|\langle X^{j,1}_{o,t}\rangle_{t\in[T]} - \frac{1}{2}\rho_o(\{i_1,i_2\})\right| \leq \sqrt{\frac{1}{2T}\ln\Big(\frac{2|J||\mathcal{O}|}{\alpha}\Big)} + E_w(1).$$

Besides, according to Assumption 8 and the definition of the sets $\bar{J}_l$, there exists p_1 such that if $j \in \bar{J}_1$ then there exists $o \in \mathcal{O}$ such that $\rho_o(\{i_1,i_2\}) \geq p_1$ and if $j \notin \bar{J}_1$ then for all $o \in \mathcal{O}$, $\rho_o(\{i_1,i_2\}) < p_1$. Let $q_1 := \max_{j_S\in J_1\setminus\bar{J}_1, o\in\mathcal{O}} \rho_o(S)$. Let us choose the threshold $s_1 := \frac{1}{2}(\frac{1}{2}p_1 + \frac{1}{2}q_1)$. Then, on $D^1 \cap D^1_{\text{sel}}$,

$$\left|\langle X^{j,1}_{o,t}\rangle_{t\in[T]} - \frac{1}{2}\rho_o(\{i_1,i_2\})\right| = O\Big(T^{-1/2} + e^{-c\sqrt{M}}\Big)$$

so there exist constants C^1_1 and C^1_2 independent of M, N and T such that for $T \geq C^1_1$ and $M \geq C^1_2$ we have for all $j_S \in \bar{J}_1$, $o \in \mathcal{O}$

$$\left|\langle X^{j,1}_{o,t}\rangle_{t\in[T]} - \frac{1}{2}\rho_o(S)\right| < \frac{1}{2}\Big(\frac{1}{2}p_1 - \frac{1}{2}q_1\Big).$$

so if $j \in \bar{J}_1$ then $\langle X^{j,1}_{o,t}\rangle_{t\in[T]} \geq s_1$ and otherwise $\langle X^{j,1}_{o,t}\rangle_{t\in[T]} < s_1$. Hence with this choice of threshold, on $D^1 \cap D^1_{\text{sel}}$ we have $\hat{J}_1 = \bar{J}_1$ and Proposition 28 is true for rank 1.

Case $l > 1$: Suppose Proposition 28 is true for rank $l-1$. Let $D^1, D^1_{\text{sel}}, \ldots, D^{l-1}$, D^{l-1}_{sel} the events and C^{l-1}_1, C^{l-1}_2 the constants given by the proposition at rank $l-1$. Suppose $\frac{T}{M^{l-2}} \geq C^{l-1}_1$ and $M \geq C^{l-1}_2$. Then the conclusions of Proposition 28 hold for rank $l-1$.

We remind that for $j = j_{S_1\cup S_2} \in J_l$ with $j_1 = j_{S_1}, j_2 = j_{S_2} \in \hat{J}_{l-1}$ and $j' \in \hat{J}_{l-1}$, the cumulated gain of neuron j' w.r.t neuron j is $G^{j'\to j}_M = \sum_{m=1}^M \langle X^{j_1,l}_{m,t} X^{j_2,l}_{m,t} X^{j',l}_{m,t}\rangle_{t\in[T]}$. Let

$$\mathring{G}^{j'\to j}_M := \sum_{m=1}^M \mathbb{E}[X^{j_1,l}_{m,1} X^{j_2,l}_{m,1} X^{j',l}_{m,1} \mid \mathcal{F}_{l-1}].$$

The variables $(X_{m,t}^{j_1,l}X_{m,t}^{j_2,l}X_{m,t}^{j',l})_{1\leq m\leq M,1\leq t\leq T}$ are independent knowing $\mathcal{F}_{l-1}$. According to Proposition 25, there exists a set D^l of probability greater than $1-\alpha$ on which for all $j\in\bar{J}_l, j'\in\hat{J}_l$,

$$|G_M^{j'\to j}-\mathring{G}_M^{j'\to j}|\leq\sqrt{\frac{M}{2T}\ln\Big(\frac{2|\hat{J}_{l-1}||J_l|}{\alpha}\Big)} \tag{19}$$

since $\mathbb{E}[X_{m,t}^{j_1,l}X_{m,t}^{j_2,l}X_{m,t}^{j',l}\mid\mathcal{F}_{l-1}]$ does not depend on t.

Let $(Z_{m,t}^{j,l})_{j\in\hat{J}_1\cup\hat{J}_{l-1},1\leq m\leq M,1\leq t\leq T}$, $(Z_{o,t}^{j,l})_{j\in\hat{J}_{l-1},o\in\mathcal{O},1\leq t\leq T}$ be the coupled network activity with arbitrary $\mathcal{F}_{l-1}$-measurable weights $(w_\star^j)_{j\in\hat{J}_1\cup\cdots\cup\hat{J}_{l-1}}$ as defined in section C.2. For $j=j_{S_1\cup S_2}\in J_l$ with $j_1=j_{S_1},j_2=j_{S_2}\in\hat{J}_{l-1}$, $j'\in\hat{J}_{l-1}$, let $\bar{G}_M^{j'\to j}:=\sum_{m=1}^M\mathbb{E}[Z_{m,1}^{j_1,l}Z_{m,1}^{j_2,l}Z_{m,1}^{j',l}\mid\mathcal{F}_{l-1}]$. Then since the variables are bounded by 1, we have

$$\begin{aligned}&|\mathring{G}_M^{j'\to j}-\bar{G}_M^{j'\to j}|\leq\\&\sum_{m=1}^M\Big(\mathbb{E}[|X_{m,t}^{j',l}-Z_{m,t}^{j',l}|\mid\mathcal{F}_{l-1}]+\mathbb{E}[|X_{m,t}^{j_1,l}-Z_{m,t}^{j_1,l}|\mid\mathcal{F}_{l-1}]+\mathbb{E}[|X_{m,t}^{j_2,l}-Z_{m,t}^{j_2,l}|\mid\mathcal{F}_{l-1}]\Big).\end{aligned} \tag{20}$$

Let $j''\in\{j',j_1,j_2\}$. Note that $j''\in\bar{J}_{l-1}$. We have

$$\begin{aligned}&\mathbb{E}[|X_{m,t}^{j'',l}-Z_{m,t}^{j'',l}|\mid\mathcal{F}_{l-1}]\leq\mathbb{E}[|p_{m,t}^{j'',l}(w_{M+1}^{j''})-p_{m,t}^{j'',l}(\bar{w}^{j''})|\mid\mathcal{F}_{l-1}]\\&\leq\mathbb{E}[|w_{M+1}^{j''}\cdot X_{m,t-1}^{\bar{J}_{l-1},l}-w_\star^{j''}\cdot Z_{m,t-1}^{\bar{J}_{l-1},l}|\mid\mathcal{F}_{l-1}]\\&\leq\mathbb{E}[|w_{M+1}^{j''}\cdot X_{m,t-1}^{\bar{J}_{l-1},l}-w_\star^{j''}\cdot X_{m,t-1}^{\bar{J}_{l-1},l}|\mid\mathcal{F}_{l-1}]+\mathbb{E}[|w_\star^{j''}\cdot X_{m,t-1}^{\bar{J}_{l-1},l}-w_\star^{j''}\cdot Z_{m,t-1}^{\bar{J}_{l-1},l}|\mid\mathcal{F}_{l-1}]\\&\leq\|w_{M+1}^{j''}-w_\star^{j''}\|_1+\max_{a\in\bar{J}_{l-2}}\mathbb{E}[|X_{m,t}^{a,l}-Z_{m,t}^{a,l}|\mid\mathcal{F}_{l-1}]\end{aligned}$$

because the variables $X_{m,t-1}^{\bar{J}_{l-1},l}=(X_{m,t-1}^{a,l})_{a\in\bar{J}_{l-1}}$ are bounded by 1 and the weights $w_\star^{j''}$ and $w_{M+1}^{j''}$ are $\mathcal{F}_{l-1}$-measurable and bounded by 1. By iteration we get

$$\mathbb{E}[|X_{m,t}^{j'',l}-Z_{m,t}^{j'',l}|\mid\mathcal{F}_{l-1}]\leq\sum_{l'=1}^{l-1}\max_{a\in\hat{J}_{l'}}\|w_{M+1}^a-w_\star^a\|_1 \tag{21}$$

Let us now work on the event $D^1\cap D_{\text{sel}}^1\cap\cdots\cap D^{l-1}\cap D_{\text{sel}}^{l-1}\cap D^l$. Then for every $l'\leq l-1$, $\hat{J}_{l'}=\bar{J}_{l'}$. For every $a\in\bar{J}_{l'}$, let us choose $w_\star^a=\bar{w}^a$. Then

$$\mathbb{E}[|X_{m,t}^{j'',l}-Z_{m,t}^{j'',l}|\mid\mathcal{F}_{l-1}]\leq\sum_{l'=1}^{l-1}E_w(l') \tag{22}$$

and we can control the $E_w(l')$. Then by combining (20) and (22) we get

$$|\mathring{G}_M^{j'\to j} - \bar{G}_M^{j'\to j}| \leq 3M \sum_{l'=1}^{l-1} E_w(l'). \tag{23}$$

For $j \in J_l, j' \in \bar{J}_{l-1}$, let $\bar{w}_{M+1}^{j'\to j} := \frac{\exp(\eta^j \bar{G}_M^{j'\to j})}{\sum_{j''\in\bar{J}_{l-1}} \exp(\eta^j \bar{G}_M^{j''\to j})}$. According to Prop. 27,

$$\begin{aligned}
&\|w_{M+1}^j - \bar{w}_{M+1}^j\|_2 \leq \eta^j \|(\bar{G}_M^{j'\to j})_{j'\in\bar{J}_{l_1}} - (G_M^{j'\to j})_{j'\in\bar{J}_{l_1}}\|_2 \\
&\leq \eta^j \Big(\|(\mathring{G}_M^{j'\to j})_{j'\in\bar{J}_{l_1}} - (G_M^{j'\to j})_{j'\in\bar{J}_{l_1}}\|_2 + \|(\bar{G}_M^{j'\to j})_{j'\in\bar{J}_{l_1}} - (\mathring{G}_M^{j'\to j})_{j'\in\bar{J}_{l_1}}\|_2 \Big)
\end{aligned}$$

and by combining (19) and (23), with $\eta^j = \sqrt{\frac{8\ln(|\bar{J}_{l-1}|)}{M}}$ we get

$$\|w_{M+1}^j - \bar{w}_{M+1}^j\|_2 \leq E_{\text{prec}}(M, |\bar{J}_{l-1}|, l-1) + E_{\text{approx}}(T, |\bar{J}_{l-1}|, |\bar{J}_{l-1}^*|, \alpha) \tag{24}$$

Let us study $\bar{G}_M^{j'\to j}$ where $j = j_{S_1\cup S_2}$. We denote $S = S_1 \cup S_2$, and for a set S_e with e an index, we denote j_e its associated neuron. For $j'' = j_{S_a\cup S_b} \in \{j', j_1, j_2\}$, the conditional spiking probability of $Z_{m,t}^{j'',l}$ is $\Big(-\frac{1}{2} + \frac{1}{2}(Z_{m,t-1}^{j_a,l} + Z_{m,t-1}^{j_b,l})\Big)_+$. Then j'' spikes with probability $\frac{1}{2}$ if and only if neurons j_a and j_b spiked before. Let $S_1 = S_{a_1} \cup S_{a_2}$, $S_2 = S_{b_1} \cup S_{b_2}$, $j' = j_{S'}$ with $S' = S_{d_1} \cup S_{d_2}$. Let $c_{l'}$ is the number of distinct variables of layer l' which intervene in the tensors j_1, j_2 and j'.

$$\begin{aligned}
\bar{G}_M^{j'\to j} &= \sum_{m=1}^M \mathbb{E}[Z_{m,t}^{j_1,l} Z_{m,t}^{j_2,l} Z_{m,t}^{j',l} \mid \mathcal{F}_{l-1}] \\
&= \sum_{m=1}^M \mathbb{E}[\frac{1}{2^{c_{l-1}^{j'\to j}}} Z_{m,t}^{j_{a_1},l} Z_{m,t}^{j_{a_2},l} Z_{m,t}^{j_{b_1},l} Z_{m,t}^{j_{b_2},l} Z_{m,t}^{j_{d_1},l} Z_{m,t}^{j_{d_2},l} \mid \mathcal{F}_{l-1}] \\
&\dots \\
&= \sum_{m=1}^M 2^{-(c_{l-1}^{j'\to j} + c_{l-2}^{j'\to j} + \dots + c_1^{j'\to j})} \mathbb{E}[\prod_{i\in S'\cup S_1\cup S_2} X_{m,t}^{i,l} \mid \mathcal{F}_{l-1}] \\
&= \sum_{m=1}^M 2^{-(c_{l-1}^{j'\to j} + c_{l-2}^{j'\to j} + \dots + c_1^{j'\to j})} \prod_{i\in S'\cup S_1\cup S_2} \mathbb{E}[X_{m,t}^{i,l} \mid \mathcal{F}_{l-1}] \\
&= \sum_{m=1}^M 2^{-(c_{l-1}^{j'\to j} + c_{l-2}^{j'\to j} + \dots + c_1^{j'\to j})} \prod_{i\in S'\cup S_1\cup S_2} \mathbb{E}[X_{m,t}^{i,l}] \\
&= \sum_{m=1}^M 2^{-(c_{l-1}^{j'\to j} + c_{l-2}^{j'\to j} + \dots + c_1^{j'\to j})} \rho_m(S_1 \cup S_2 \cup S')
\end{aligned}$$

because the variables $X_{m,t}^{i,l}$ are mutually independent knowing $\mathcal{F}_{l-1}$ and are independent of $\mathcal{F}_{l-1}$. Therefore, according to Assumption 5,

$$\bar{G}_M^{j'\to j} = 2^{-(c_{l-1}^{j'\to j}+c_{l-2}^{j'\to j}+\cdots+c_1^{j'\to j})}\rho(S'\cup S)M.$$

There are two cases. If $\rho(S)=0$ then for all $j'\in\bar{J}_{l-1}$, $\bar{G}_M^{j'\to j}=0$. Let $\bar{w}^{j'\to j}:=\frac{1}{|\bar{J}_{l-1}|}$. This definition does not conflict with the one recalled at the beginning of the proof because $j\notin\bar{J}_l$ according to the definition of $\bar{J}_l$ given in section 3.1.

Then for all $j'\in\bar{J}_{l-1}$,

$$\bar{w}_{M+1}^{j'\to j} = \bar{w}^{j'\to j}. \tag{25}$$

If $\rho(S)>0$ then according to the definition of J_l, $S_1\cap S_2=\emptyset$ so for $j'\in\{j_1,j_2\}$, $c_{l'}^{j'\to j}=2^{l-l'}$ for any $l'\le l-1$. So $\bar{G}_M^{j'\to j}=2^{-2^l+2}\rho(S)M$. If $j'\notin\{j_1,j_2\}$ then $c_{l-1}^{j'\to j}=3$, otherwise $c_{l-1}^{j'\to j}=2$. Besides, it is clear that if $j'\in\{j_1,j_2\}$ and $j''\notin\{j_1,j_2\}$ then for every $l'\le l-2$, $c_{l'}^{j'\to j}\le c_{l'}^{j''\to j}$. In addition, $\rho(S)\ge\rho(S'\cup S)$ for any $j'\in\bar{J}_{l-1}$. Therefore for $j'\neq j_1,j_2$, $\bar{G}_M^{j'\to j}\le 2^{-2^l+1}\rho(S)M$. Hence

$$\arg\max_{j'\in\bar{J}_{l-1}} 2^{-(c_{l-1}^{j'\to j}+c_{l-2}^{j'\to j}+\cdots+c_1^{j'\to j})}\rho(S'\cup S)M = \{j_1,j_2\}$$

and

$$\max_{j'\in\bar{J}_{l-1}} 2^{-(c_{l-1}^{j'\to j}+c_{l-2}^{j'\to j}+\cdots+c_1^{j'\to j})}\rho(S'\cup S) - \max_{j'\neq j_1,j_2} 2^{-(c_{l-1}^{j'\to j}+c_{l-2}^{j'\to j}+\cdots+c_1^{j'\to j})}\rho(S'\cup S) \ge 2^{-2^l+1}\rho(S).$$

Let $\bar{w}^{j'\to j}:=\mathbb{1}_{j'\in\{j_1,j_2\}}$. According to Proposition 26, with $\eta^j=\sqrt{\frac{8\ln(|\bar{J}_{l-1}|)}{M}}$ we have for all $j'\in\bar{J}_{l-1}$

$$|\bar{w}_{M+1}^{j'\to j}-\bar{w}^{j'\to j}| \le E_{\mathrm{EWA}}^{j,l}(M,|\bar{J}_{l-1}|). \tag{26}$$

Therefore, for $j=j_S\in J_l$,

- if $\rho(S)=0$ then by combining (25) and (24) we get

$$\|w_{M+1}^j-\bar{w}^j\|_2 \le E_{\mathrm{prec}}(M,|\bar{J}_{l-1}|,l-1)+E_{\mathrm{approx}}(T,|\bar{J}_{l-1}|,|\bar{J}_{l-1}^*|,\alpha).$$

- if $\rho(S)>0$ then by combining (24) and (26) we get

$$\|w_{M+1}^j-\bar{w}^j\|_2 \le E_{\mathrm{prec}}(M,|\bar{J}_{l-1}|,l-1)+E_{\mathrm{approx}}(T,|\bar{J}_{l-1}|,|\bar{J}_{l-1}^*|,\alpha)+\sqrt{|\hat{J}_{l-1}|}E_{\mathrm{EWA}}^{j,l}(M,|\bar{J}_{l-1}|).$$

Now let us study the selection phase. Let $\mathcal{F}_l^{\text{learn}} := \mathcal{F}_{M,T}^l$ be the σ-algebra generated by every event that happened until the end of the learning phase of layer l. Let $j \in J_l$. The variables $(X_{o,t}^{j,l})_{o\in\mathcal{O},1\leq t\leq T}$ are independent knowing the σ-algebra $\mathcal{F}_l^{\text{learn}}$ since the weights w_{M+1}^j are frozen and are $\mathcal{F}_l^{\text{learn}}$-measurable. Hence according to Proposition 25, there exists a set D_{sel}^l of probability more than $1-\alpha$ on which for every $j \in J_l$, $o \in \mathcal{O}$,

$$\left|\langle X_{o,t}^{j,l}\rangle_{t\in[T]} - \mathbb{E}[X_{o,1}^{j,l} \mid \mathcal{F}_l^{\text{learn}}]\right| \leq \sqrt{\frac{1}{2T}\ln\Big(\frac{2|J_l||\mathcal{O}|}{\alpha}\Big)}$$

since $\mathbb{E}[X_{o,t}^{j,l} \mid \mathcal{F}_l^{\text{learn}}]$ does not depend on t.

Let $(Z_{o,t}^{j,l})_{j\in\hat{J}_1\cup\cdots\cup\hat{J}_{l-1}\cup J_l, o\in\mathcal{O}, 1\leq t\leq T}$ be the coupled network activity with any weights $(w_\star^j)$ which are $\mathcal{F}_l^{\text{learn}}$-measurable as defined in section C.2. Then for $j = j_S \in J_l$, $o \in \mathcal{O}$, similarly as in (21) we get

$$\mathbb{E}[|X_{o,1}^{j,l} - Z_{o,1}^{j,l}| \mid \mathcal{F}_l^{\text{learn}}] \leq \sum_{l'=1}^{l-1} \max_{a\in\hat{J}_{l'}} \|w_{M+1}^a - w_\star^a\|_1 + \max_{a\in J_l} \|w_{M+1}^a - w_\star^a\|_1.$$

Let us work on $D^1 \cap D_{\text{sel}}^1 \cap \cdots \cap D^l \cap D_{\text{sel}}^l$. Let $w_\star^j := \bar{w}^j$ for $j \in \bar{J}_1 \cup \cdots \cup \bar{J}_{l-1} \cup J_l$. Then we by Cauchy-Schwartz inequality we have

$$\left|\langle X_{o,t}^{j,l}\rangle_{t\in[T]} - \mathbb{E}[Z_{o,1}^{j,l} \mid \mathcal{F}_l^{\text{learn}}]\right| \leq \sqrt{\frac{1}{2T}\ln\Big(\frac{2|J_l||\mathcal{O}|}{\alpha}\Big)} + \sum_{l'=1}^{l} E_w(l') + \sqrt{|\hat{J}_{l'-1}|}\max_{a\in J_l}\|w_{M+1}^a - \bar{w}^a\|_2$$

where we control every term. Let us study $\mathbb{E}[Z_{o,1}^{j,l} \mid \mathcal{F}_l^{\text{learn}}]$. If $\rho(S) > 0$ then similarly as in the computation of $\bar{G}_M^{j'\to j}$ we get that $\mathbb{E}[Z_{o,1}^{j,l} \mid \mathcal{F}_l^{\text{learn}}] = 2^{-2^l+1}\rho(S)$. If $\rho(S) = 0$ then

$$\mathbb{E}[Z_{o,t}^{j,l} \mid \mathcal{F}_l^{\text{learn}}] = \mathbb{E}\Big[\Big(-\frac{1}{2} + \frac{1}{|\bar{J}_{l-1}|}\sum_{j'\in\bar{J}_{l-1}} Z_{o,t-1}^{j',l}\Big)_+ \mid \mathcal{F}_l^{\text{learn}}\Big].$$

The computation in the case $\rho(S) > 0$ also holds for neurons of layer $l-1$, which are all in this case because $\hat{J}_{l-1} = \bar{J}_{l-1}$. Therefore, according to Assumption 8, at most half the neurons of layer $l-1$ are active when presented with o. Hence

$$\mathbb{E}[Z_{o,1}^{j,l} \mid \mathcal{F}_l^{\text{learn}}] = 0 = 2^{-2^l+1}\rho(S).$$

Therefore for all $j \in J_l$, $o \in \mathcal{O}$,

$$\left|\langle X_{o,t}^{j,l}\rangle_{t\in[T]} - 2^{-2^l+1}\rho(S)\right|$$

$$\leq \sqrt{\frac{1}{2T}\ln\Big(\frac{2|J_l||\mathcal{O}|}{\alpha}\Big)} + \sum_{l'=1}^{l} E_w(l') + \sqrt{|\hat{J}_{l'-1}|}\max_{a\in J_l}\|w^a_{M+1} - \bar{w}^a\|_2 := E(l,M,T)$$

Besides, by definition of $\bar{J}_l$, if $j \in \bar{J}_l$ then there exists $o \in \mathcal{O}$ such that $\rho_o(S) \geq 2^{2^l-1}s_l$ and if $j \in J_l \setminus \bar{J}_l$ then for all $o \in \mathcal{O}$, $\rho_o(S) < 2^{2^l-1}s_l$. Let $q_l := 2^{-2^l+1}\max_{j_S\in J_l\setminus\bar{J}_l}\rho_o(S)$. Then $q_l < s_l$. Since $E(l,M,T) = O\Big((\frac{M^{l-1}}{T})^{1/2} + e^{-C\sqrt{M}}\Big)$ then for $\frac{T}{M^{l-1}}$ and M large enough,

$$E(l,M,T) < s_l - q_l. \tag{27}$$

Hence there exists constants C_1^l and C_2^l such that for $\frac{T}{M^{l-1}} \geq C_1^l$ and $M \geq C_2^l$, (27) is true. Besides, $\frac{T}{M^{l-2}} \geq \frac{T}{M^{l-1}}$ so let us take $C_1^l \geq C_1^{l-1}$ and $C_2^l \geq C_2^{l-1}$ so that we still have $\frac{T}{M^{l-2}} \geq C_1^{l-1}$ and $M \geq C_2^{l-1}$. Under this condition and this choice of threshold, if $j \in \bar{J}_l$ then $\langle X^{j,l}_{o,t}\rangle_{t\in[T]} \geq s_l$ and otherwise $\langle X^{j,l}_{o,t}\rangle_{t\in[T]} < s_l$ so $\hat{J}_l = \bar{J}_l$ and Proposition 28 is true for rank l.

Now that Proposition 28 is proved for every $l \in [L]$, we can directly deduce Theorem 9 by bounding $|J_l|$ by $|\bar{J}_{l-1}|^2$.

C.4 Proof of Corollary 10

Let $l \in [L]$, $j \in \bar{J}_l$, $o \in \mathcal{O}$. With the same calculation as in the proof of Theorem 9 (see section C.3), we have $\mathbb{E}[p^j_o(\bar{w}^j)] = 2^{-2^l+1}\rho(S)$. Besides, for T, M and T/M^{L-1} tending to infinity, the error term of Theorem 9 tends to zero so for T, M and T/M^{L-1} large enough, for every $l \in [L]$, the set of selected neurons is always $\bar{J}_l$ and the limit layer is an ideal hidden layer with constant 2^{-2^l+1}.

C.5 Proof of Corollary 12

Suppose Assumption 11 and the assumptions of Theorem 9 hold. Let $\alpha > 0$. Suppose $T/M^{L-1} \geq C_1$ and $M \geq C_2$ where C_1 and C_2 are the constants defined in Theorem 9 and $\tilde{\mathcal{Q}}_L$ is non-empty. Then the conclusions of Theorem 9 hold. Let us use the notations introduced in the proof of Theorem 9.

Let $k \in K, j \in \hat{J}_L$. Let us define $S_N^{j\to k} := \sum_{m,o_m^{L+1}\in k}\langle X^{j,L+1}_{m,t}\rangle_{t\in[T]}$ and $\mathring{S}_N^{j\to k} := \sum_{m,o_m^{L+1}\in k}\mathbb{E}[X^{j,L+1}_{m,1} \mid \mathcal{F}_L]$. The variables $(X^{j,L+1}_{m,t})_{1\leq t\leq T, m \text{ s.t. } o_m^{L+1}\in k}$ are bounded by 1 and independent knowing $\mathcal{F}_L$. Hence according to Proposition 25, there exists an event D^{L+1} of probability more than $1-\alpha$ such that on D^{L+1}, for every $k \in K$, $j \in \hat{J}_L$ we have

$$|S_N^{j\to k} - \mathring{S}_N^{j\to k}| \leq \sqrt{\frac{N^k}{2T}\ln\Big(\frac{2|\hat{J}_L||K|}{\alpha}\Big)}.$$

Let $(Z_{m,t}^{j,L+1})_{j\in\hat{J}_1\cup\cdots\cup\hat{J}_L,1\leq m\leq M,1\leq t\leq T}$ be the coupled network activity with arbitrary weights $(w_\star^j)_{j\in\hat{J}_1\cup\cdots\cup\hat{J}_L}$ which are $\mathcal{F}_L$-measurable as defined in section C.2. For $k\in K, j\in\hat{J}_L$ let

$$\bar{S}_N^{j\to k}:=\sum_{m,o_m^{L+1}\in k}\mathbb{E}[Z_{m,1}^{j,L+1}\mid\mathcal{F}_L].$$

Then,

$$|\bar{S}_N^{j\to k}-\mathring{S}_N^{j\to k}|\leq\sum_{m,o_m^{L+1}\in k}\mathbb{E}[|X_{m,1}^{j,L+1}-Z_{m,1}^{j,L+1}|\mid\mathcal{F}_L]$$

and similarly as in the proof of Theorem 9 (see section C.3), we have

$$\begin{aligned}|\bar{S}_N^{j\to k}-\mathring{S}_N^{j\to k}|&\leq\sum_{m,o_m^{L+1}\in k}\sum_{l=1}^{L}\sqrt{|\hat{J}_l|}\max_{a\in\hat{J}_l}\|w_{M+1}^a-w_\star^a\|_2\\&=N^k\sum_{l=1}^{L}\sqrt{|\hat{J}_l|}\max_{a\in\hat{J}_l}\|w_{M+1}^a-w_\star^a\|_2.\end{aligned}$$

Let us work on $D:=D^1\cap D_{\text{sel}}^1\cap\cdots\cap D^L\cap D_{\text{sel}}^L\cap D^{L+1}$ where for $l\in[L]$, D^l and D_{sel}^l are the events given in the proof of Theorem 9 (see section C.3). Then D is of probability greater than $1-2(L+1)\alpha$. Choose $w_\star=\bar{w}$. Then, for $j\in\bar{J}_L$, $k\in K$,

$$|S_N^{j\to k}-\bar{S}_N^{j\to k}|\leq\sqrt{\frac{N^k}{2T}\ln\Big(\frac{2|\bar{J}_L||K|}{\alpha}\Big)}+N^k\sum_{l=1}^{L}E_w(l).\tag{28}$$

Here, according to Assumption 5, every nature of object is presented the same amount of time to the network during the learning phase of the output layer. Hence Assumption 23 holds for $\xi=\frac{1}{|\mathcal{O}|}$. Therefore, according to Proposition 24,

$$\mathrm{Disc}_N(w_{[N]}^k,X_{[N],[T]}^{\hat{J}_L,L+1})\geq\max_{q^K\in(\mathcal{P}_{\bar{J}_L})^{|K|}}\mathrm{Disc}_N(q^K,X_{[N],[T]}^{\hat{J}_L,L+1})-\frac{C|\mathcal{O}||K|}{|K|-1}\sqrt{\frac{\ln(|\hat{J}_L|)}{N}}.$$

Let $q^K\in(\mathcal{P}_{\bar{J}_L})^{|K|}$ be a feasible weight family.

$$\begin{aligned}\mathrm{Disc}_N(q^K,X_{[N],[T]}^{\hat{J}_L,L+1})&=\Big\langle\hat{p}_m^k(q^k)-\hat{p}_m^{k'}(q^{k'})\Big\rangle_{\substack{k\in K\\k'\neq k\\m,o_m^{L+1}\in k}}\\&=\Big\langle(q^k-q^{k'})\cdot\Big(\frac{1}{N^k}S_N^{j\to k}\Big)_{j\in\bar{J}_L}\Big\rangle_{\substack{k\in K\\k'\neq k}}.\end{aligned}$$

Besides, similarly as in the proof of Theorem 9, for $j = j_S$ we have $\mathbb{E}[X_{m,1}^{j,L+1} \mid \mathcal{F}_L] = 2^{-2^L+1}\rho_{o_m^{L+1}}(S)$ so

$$\Big\langle (q^k - q^{k'}) \cdot \Big(\frac{1}{N^k}\bar{S}_N^{j\to k}\Big)_{j\in \bar{J}_L}\Big\rangle_{\substack{k\in K\\ k'\neq k}} = 2^{-2^L+1}\Big\langle (q^k - q^{k'}) \cdot \rho_o(S)\Big\rangle_{\substack{k\in K\\ k'\neq k\\ o\in k}}$$
$$= \gamma_L \operatorname{Disc}^{\mathrm{id}}(q^K, \bar{J}_L)$$

with $\gamma_L = 2^{-2^L+1}$. Therefore, by lower bounding N^k by $\frac{N}{|\mathcal{O}|}$ for every k we get

$$\operatorname{Disc}_N(q^K, X_{[N],[T]}^{\hat{J}_L,L+1}) \geq \gamma_L \operatorname{Disc}^{\mathrm{id}}(q^K, \bar{J}_L) - 2\sqrt{\frac{|\mathcal{O}|}{2TN}\ln\Big(\frac{2|\bar{J}_L||K|}{\alpha}\Big)} - 2\sum_{l=1}^{L} E_w(l).$$

Hence

$$\operatorname{Disc}_N(w_{[N]}^k, X_{[N],[T]}^{\hat{J}_L,L+1}) \geq \gamma_L \operatorname{Disc}^{\mathrm{id}}(q^K, \bar{J}_L) - 2\sqrt{\frac{|\mathcal{O}|}{2TN}\ln\Big(\frac{2|\bar{J}_L||K|}{\alpha}\Big)}$$
$$- 2\sum_{l=1}^{L} E_w(l) - \frac{C|\mathcal{O}||K|}{|K|-1}\sqrt{\frac{\ln(|\hat{J}_L|)}{N}}.$$

C.6 Proof of Theorem 14

Suppose Assumption 11 and the assumptions of Theorem 9 hold. Let $\alpha > 0$. Suppose $T/M^{L-1} \geq C_1$ and $M \geq C_2$ where C_1 and C_2 are the constants defined in Theorem 9. Then the conclusions of Theorem 9 hold. For $k \in K$, recall that $(\bar{w}^{j\to k})_{j\in\bar{J}_L} = \frac{1}{|\bar{J}_L^k|}\mathbb{1}_{|\bar{J}_L^k|}$. Let us work on the event D defined in the proof of Corollary 12 and let us use the notations introduced in section C.5. Then $G_N^{j\to k} = \frac{N}{N^k}S_N^k - \Big\langle \frac{N}{N^{k'}}S_N^{k'}\Big\rangle_{k'\neq k}$. Let $\bar{G}_N^{j\to k} = \frac{N}{N^k}\bar{S}_N^k - \Big\langle \frac{N}{N^{k'}}\bar{S}_N^{k'}\Big\rangle_{k'\neq k}$. Then according to (28),

$$|G_N^{j\to k} - \bar{G}_N^{j\to k}| \leq N\sqrt{\frac{1}{2TN^k}\ln\Big(\frac{2|\bar{J}_L||K|}{\alpha}\Big)} + \Big\langle N\sqrt{\frac{1}{2TN^{k'}}\ln\Big(\frac{2|\bar{J}_L||K|}{\alpha}\Big)}\Big\rangle_{k'\neq k}$$
$$+ 2N\sum_{l=1}^{L} E_w(l).$$

Let us study $\bar{G}_N^{j\to k}$. Similarly as in the proof of Theorem 9 (see section C.3), for $j = j_S$

$$\mathbb{E}[Z_{m,1}^{j,L+1} \mid \mathcal{F}_L] = 2^{-2^L+1}\rho_{o_m^{L+1}}(S). \tag{29}$$

Then $\bar{G}_N^{j\to k} = 2^{-2^L+1}\operatorname{Disc}^{j\to k}$. Let $\bar{w}_{N+1}^{j\to k} := \frac{\exp(\eta^{L+1}\bar{G}_N^{j\to k})}{\sum_{h\in\bar{J}_L}\exp(\eta^{L+1}\bar{G}_N^{h\to k})}$. Let $k \in K$, $E^k := \arg\max_{j\in\bar{J}_L}\operatorname{Disc}^{j\to k}$. Then according to Proposition 26, if $E^k = \bar{J}_L$ then

for every $j \in \bar{J}_L$ we have $\bar{w}_{N+1}^{j\to k} = \bar{w}^{j\to k}$. Otherwise,

$$|\bar{w}_{N+1}^{j\to k} - \bar{w}^{j\to k}| \leq E_{\mathrm{EWA}}^k(N, |\bar{J}_L|)$$

where

$$E_{\mathrm{EWA}}^k(N, |\bar{J}_L|) := \frac{1}{|\bar{J}_L|} \max\Big(1, \frac{|\bar{J}_L| - |\bar{J}_L^k|}{|\bar{J}_L^k|}\Big) \exp\Big(- 2^{-2^L+1} \Delta^k \sqrt{8N \ln(|\bar{J}_L|)}\Big).$$

Let $\xi(N, |\bar{J}_L|, |O|, T, K) = \sqrt{\frac{\ln(|\bar{J}_L|)}{|\mathcal{O}|T} \ln\Big(\frac{2|\bar{J}_L||K|}{\alpha}\Big)} + \frac{1}{|\mathcal{O}|}\sqrt{8|\bar{J}_L| \ln(|\bar{J}_L|)N} \sum_{l=1}^L E_w(l)$.
Besides, according to Proposition 27, with $\eta^{L+1} = \frac{1}{|\mathcal{O}|}\sqrt{\frac{2\ln(|\bar{J}_L|)}{N}}$, by bounding every $\frac{N}{N^{k'}}$ by $|\mathcal{O}|$ we get that $\|w_{N+1}^k - \bar{w}_{N+1}^k\|_2 \leq \xi(N, |\bar{J}_L|, |O|, T, K)$. Hence,

$$\|w_{N+1}^k - \bar{w}^k\|_2 \leq \begin{cases} \xi(N, |\bar{J}_L|, |O|, T, K) & \text{if } E^k = \bar{J}_L, \\ \xi(N, |\bar{J}_L|, |O|, T, K) + \sqrt{|\bar{J}_l|} E_{\mathrm{EWA}}^k(N, |\bar{J}_L|) & \text{otherwise.} \end{cases}$$

C.7 Proof of Corollary 15

When T, M, N and $\frac{T}{NM^{L-1}}$ tend to infinity, the error terms of Theorem 14 tend to zero so we have the result.

C.8 Proof of Theorem 18

Suppose Assumption 16 (binary correlations) is verified. We denote $(i), (ii)$ and (iii) the three points of the theorem.
Proof of $(i) \Rightarrow (iii)$. Suppose Assumption 17 (class decomposition) is verified. Let us compute the feature discrepancies of the neurons of $\bar{J}_L$ in order to compute the weights $\bar{w}^K$ given in Corollary 15. For every $k \in K$, we choose the set E^k given in Assumption 17 (class decomposition) with maximal size. Let $j = j_S \in \bar{J}_L$.

$$\mathrm{Disc}^{j\to k} = \rho^k(S) - \langle \rho^{k'}(S)\rangle_{k'\neq k} = \langle \rho_o(S)\rangle_{o\in k} - \langle \rho_o(S)\rangle_{k'\neq k, o\in k'}.$$

Then according to Assumption 16 (binary correlations), $\rho_o(S) = p$ if and only if o has features S, *i.e.,* if and only if $o \in \mathcal{O}^S$, and $\rho_o(S) = 0$ otherwise.

Let $j = j_S$ where $S \in E^k$. Let $o \in k$. We have $\langle \rho_o(S)\rangle_{o\in k} = \frac{|\mathcal{O}^S|}{n^k} p = \frac{Cp}{n^k}$. Let $o \in k' \neq k$. If $\rho_o(S) > 0$, it would mean that o has features S so o would belong to O^S, then o would belong to class k. This is impossible so $\rho_o(S) = 0$. Therefore, $\mathrm{Disc}^{j\to k} = \frac{Cp}{n^k}$. Let $j = j_S$ where $S \notin E^k$. This means that there exists $o \in \mathcal{O}^S$ such that $o \notin k$. Therefore $\langle \rho_o(S)\rangle_{o\in k} \leq \frac{(C-1)p}{n^k}$ and $\mathrm{Disc}^{j\to k} \leq \frac{(C-1)p}{n^k}$. Hence $\bar{J}^k = E^k$ (where $\bar{J}^k$ is the set defined in section 3.1 and we identify each neurons j_S with its set S). So according to Corollary 15, for every $k \in K$ $\bar{w}^K = \frac{1}{|E^k|}\mathbb{1}_{j\in E^k}$.

Let $\bar{J}_L$ an ideal hidden layer where we denote by $(p_o^j)_{o\in\mathcal{O},j\in\bar{J}_L}$ the neurons spiking probabilities. Let γ_L its constant. Let $k \in K$, $o \in \mathcal{O}$. The spiking probability of neuron k when presented with object o is

$$p_o^k(\bar{w}^k) = \bar{w}^K \cdot p_o^{\bar{J}_L} = \gamma_L \bar{w}^K \cdot (\rho_o(S))_{j_S\in\bar{J}_L} = \frac{\gamma_L}{|E^k|} \sum_{j_S\in E^k} \rho_o(S) = \frac{p\gamma_L n_o^k}{|E^k|},$$

where n_o^k is the amount of sets O^S containing o for $j_S \in E^k$. If $o \in k$, according to Assumption 17 (class decomposition) $n_o^k \geq 1$. If $o \notin k$ then $n_o^k = 0$ (otherwise o would belong to a set O^S in the composition of class k so o would belong to k). Therefore, $p_o^k(\bar{w}^k) > 0$ if, and only if, $o \in k$ so $\bar{w}^K$ is a strong feasible weight family for the ideal layer $\bar{J}_L$.

Proof of $(ii) \Rightarrow (i)$. Suppose there exists a strong feasible weight family q^K w.r.t. $\bar{J}_L$. Let $k \in K$ and $E^k := \{S \subset I_L \text{ such that } j_S \in \bar{J}_L \text{ and } q^{j\to k} > 0\}$. Let us show that $k = \bigcup_{S\in E^k} \mathcal{O}^S$. The ideal spiking probability of neuron k when presented with $o \in \mathcal{O}$ is $\bar{p}_o^k(q^k, \bar{J}_L) = \sum_{j_S\in\bar{J}_L} q^{j\to k}\rho_o(S)$. Let $o \in k$. Then $\bar{p}_o^k(q^k, \bar{J}_L) > 0$ so there exists $j = j_S \in \bar{J}_L$ such that $q^{j\to k} > 0$ and $\rho_o(S) > 0$. Hence $j \in E^k$ and o has features S so $o \in \mathcal{O}^S$. Therefore $k \subset \bigcup_{S\in E^k} \mathcal{O}^S$. Let $o \in \bigcup_{S\in E^k} \mathcal{O}^S$. There exists $j = j_S \in E^k$ such that $o \in \mathcal{O}^S$ and $q^{j\to k} > 0$. Then $\bar{p}_o^k(q^k, \bar{J}_L) > 0$ so $o \in k$. Therefore, $\bigcup_{S\in E^k} \mathcal{O}^S \subset k$. It is obvious that $(iii) \Rightarrow (ii)$ so we can conclude.

C.9 Proof of Theorem 3

Under the assumptions of Theorem 3, the conclusions of Corollaries 10 and 15 and Theorem 18 hold: by combining them, we get the conclusions of Theorem 3.

C.10 Proof of Proposition 19

Let $j = j_S \in J_L$. Let $o^j := (o^{i\to j})_{i\in I}$ where $o^{i\to j} = 1$ if and only if $i \in S$. Then j is the only neuron of layer $\bar{J}_L$ activated by o^j. Indeed, all the sets S' for $j_{S'} \in J_L$ have the same size and only a neuron $j_{S'}$ with smaller $|S'|$ can also be activated by o^j. Let $E := \{o^j, j \in \bar{J}_L\}$. Then $|E| = |\bar{J}_L|$ because all the o^j are distinct. This set can be fully shattered by H: indeed, let $E' \subset E$, and $F := \{j \in \bar{J}_L, \exists o \in E', o = o^j\}$. Then E' and $E \setminus E'$ are separated by $\mathbb{1}_F$.

Let E be a set of objects that can be fully shattered by H. In particular, for each $o \in E$, there exists $F \subset \bar{J}_L$ non empty such that such that o is the unique object in E which activates a neuron in F. Let j_o be such a neuron. Then $o \mapsto j_o$ is an injection so $|E| \leq |\bar{J}_L|$.

C.11 Proof of proposition 20

Let us use the notations of Theorem 14. Let $j \in \bar{J}_L$, $k \in K$. The limit cumulative gain in M and T of a hidden neuron is of the form

$$\bar{G}_N^{j\to k} := CN\Big(\langle p^j_{o,\infty}\rangle_{o\in k} - \langle p^j_{o,\infty}\rangle_{\substack{k' \ / \ k'\neq k\\ o\in k'}}\Big)$$

where $C > 0$ is a constant and $p^j_{o,\infty}$ is the limit spiking probability of hidden neuron j when presented with object of nature o. Since limit hidden layers are not ideal layers, another quantity replaces the feature discrepancy: the quantity

$$c^{j\to k} := \langle p^j_{o,\infty}\rangle_{o\in k} - \langle p^j_{o,\infty}\rangle_{\substack{k' \ / \ k'\neq k\\ o\in k'}}$$

and according to Proposition 26, the output weights of neuron $k \in K$ converge to uniform distribution of weight on the set $\arg\max_{j\in\bar{J}_L} c^{j\to k}$. In order to compute the limit weights of neuron k_2, let us compute the quantities $c^{j\to k_2}$. Let $m \in [N]$. The limit conditional spiking probability of a hidden neuron $j = j_{\{i_1,i_2\}}$ is $\Big(-\nu + \frac{1}{2}(X^{i_1}_{o,t} + X^{i_2}_{o,t})\Big)_+$. Therefore, when only one of the two input neurons i_1 and i_2 is active, the spiking probability of j is $q := \Big(\frac{1}{2} - \nu\Big)p'$ and when the two neurons i_2 and i_2 are active, the spiking probability of j is $p := p'^2(1-\nu) + 2p'(1-p')(\frac{1}{2}-\nu)$. By definition of the feature discrepancy we have

$$\begin{aligned}
d^{j_{\{\text{Blue,Square}\}}\to k_2} &= \frac{1}{5}p - q\\
d^{j_{\{\text{Red,Circle}\}}\to k_2} &= \frac{1}{5}p + \frac{2}{5}q - \frac{1}{2}q = \frac{1}{5}p - \frac{1}{10}q\\
d^{j_{\{\text{Blue,Circle}\}}\to k_2} &= \frac{3}{5}q - \frac{1}{4}p - \frac{1}{4}q = \frac{7}{20}q - \frac{1}{4}p
\end{aligned}$$

By symmetry, we know every $d^{i\to k_2}$. Since $\nu < \frac{1}{2}$ we have $q > 0$. Then $d^{j_{\{\text{Blue,Square}\}}\to k_2} < d^{j_{\{\text{Red,Circle}\}}\to k_2}$. Besides, $d^{j_{\{\text{Red,Circle}\}}\to k_2} - d^{j_{\{\text{Blue,Circle}\}}\to k_2} = \frac{9}{20}p - \frac{9}{20}q > 0$, because $p > q$. Therefore, the limit weights of k_2 converge to the family putting weight $1/4$ on $j_{\{\text{Red,Circle}\}}$, $j_{\{\text{Red,Triangle}\}}$, $j_{\{\text{Green,Circle}\}}$ and $j_{\{\text{Green,Triangle}\}}$. Hence, the limit activity of k_2 is equal to zero when presented with $\square$, so this objects is not well classified.

References

Emmanuel Bacry, Iacopo Mastromatteo, and Jean-François Muzy. Hawkes processes in finance. *Market Microstructure and Liquidity*, 1(01):1550005, 2015.

Emmanuel Bacry, Martin Bompaire, Philip Deegan, Stéphane Gaïffas, and Søren V Poulsen. tick: A python library for statistical learning, with an emphasis on hawkes processes and time-dependent models. *JMLR*, 18(1):7937–7941, 2017.

Yujia Bao, Zhaobin Kuang, Peggy Peissig, David Page, and Rebecca Willett. Hawkes process modeling of adverse drug reactions with longitudinal observational data. In *Proceedings of the 2nd Machine Learning for Healthcare Conference*, volume 68, pages 177–190, 2017.

Alberto Bietti, Vivien Cabannes, Diane Bouchacourt, Herve Jegou, and Leon Bottou. Birth of a transformer: A memory viewpoint. In *NeurIPS*, 2023.

Pierre Brémaud and Laurent Massoulié. Stability of nonlinear hawkes processes. *Ann. Prob.*, pages 1563–1588, 1996.

Natalia Caporale and Yang Dan. Spike timing–dependent plasticity: a hebbian learning rule. *Annu. Rev. Neurosci.*, 31:25–46, 2008.

Nicolo Cesa-Bianchi and Gábor Lugosi. *Prediction, learning, and games*. Cambridge university press, 2006.

Wen-Hao Chiang and George Mohler. Hawkes process multi-armed bandits for disaster search and rescue. *arXiv preprint arXiv:2004.01580*, 2020.

Jean-Baptiste Cordonnier, Andreas Loukas, and Martin Jaggi. On the relationship between self-attention and convolutional layers. In *ICLR*, 2020.

Marla Feller and Daniel Kerschensteiner. Retinal waves and their role in visual system development. In *Synapse development and maturation*, pages 367–382. Elsevier, 2020.

Nicolas Frémaux and Wulfram Gerstner. Neuromodulated spike-timing-dependent plasticity, and theory of three-factor learning rules. *Frontiers in neural circuits*, 9:85, 2016.

Kunihiko Fukushima and Sei Miyake. Neocognitron: A new algorithm for pattern recognition tolerant of deformations and shifts in position. *Pattern Recognit.*, 15(6):455–469, 1982.

Antonio Galves and Eva Löcherbach. Modeling networks of spiking neurons as interacting processes with memory of variable length. *J. Soci. Fr. Stat.*, 157(1): 17–32, 2016.

Bolin Gao and Lacra Pavel. On the properties of the softmax function with application in game theory and reinforcement learning, 2018.

Felipe Gerhard, Moritz Deger, and Wilson Truccolo. On the stability and dynamics of stochastic spiking neuron models: Nonlinear hawkes process and point process glms. *PLoS computational biology*, 13(2):e1005390, 2017.

George L Gerstein, Purvis Bedenbaugh, and Ad MHJ Aertsen. Neuronal assemblies. *IEEE Transactions on Biomedical Engineering*, 36(1):4–14, 1989.

Mark A Gluck and Gordon H Bower. From conditioning to category learning: an adaptive network model. *J. Exp. Psychol.*, 117(3):227, 1988.

Eric C Hall and Rebecca M Willett. Tracking dynamic point processes on networks. *IEEE Trans. Inf. Theory*, 62(7):4327–4346, 2016.

Alan G Hawkes. Spectra of some self-exciting and mutually exciting point processes. *Biometrika*, 58(1):83–90, 1971.

Alan G Hawkes. Hawkes processes and their applications to finance: a review. *Quantitative Finance*, 18(2):193–198, 2018.

Donald Olding Hebb. *The organization of behavior: A neuropsychological theory*. Psychology press, 2005.

Robert F Hevner. Development of connections in the human visual system during fetal mid-gestation: a dil-tracing study. *Journal of Neuropathology & Experimental Neurology*, 59(5):385–392, 2000.

Pierre Hodara and Eva Löcherbach. Hawkes processes with variable length memory and an infinite number of components. *Adv. Appl. Probab.*, 49(1):84–107, 2017.

David H Hubel and Torsten N Wiesel. Receptive fields, binocular interaction and functional architecture in the cat's visual cortex. *J. Physiol.*, 160(1):106, 1962.

Sophie Jaffard, Samuel Vaiter, Alexandre Muzy, and Patricia Reynaud-Bouret. Provable local learning rule by expert aggregation for a Hawkes network. In *AISTATS*, 2024.

Matthias Kirchner. An estimation procedure for the hawkes process. *Quantitative Finance*, 17(4):571–595, 2017.

John K Kruschke. Alcove: An exemplar-based connectionist model of category learning. In *Connectionist psychology: a text with readings*, pages 107–138. Psychology Press, 2020.

Regis C Lambert, Christine Tuleau-Malot, Thomas Bessaih, Vincent Rivoirard, Yann Bouret, Nathalie Leresche, and Patricia Reynaud-Bouret. Reconstructing the functional connectivity of multiple spike trains using hawkes models. *J. Neurosci. Methods*, 297:9–21, 2018.

Yann LeCun, Bernhard Boser, John Denker, Donnie Henderson, Richard Howard, Wayne Hubbard, and Lawrence Jackel. Handwritten digit recognition with a back-propagation network. In *NeurIPS*, 1989.

Robert Legenstein, Christian Naeger, and Wolfgang Maass. What can a neuron learn with spike-timing-dependent plasticity? *Neural Comput.*, 17(11):2337–2382, 2005.

Robert Legenstein, Dejan Pecevski, and Wolfgang Maass. A learning theory for reward-modulated spike-timing-dependent plasticity with application to biofeedback. *PLoS computational biology*, 4(10):e1000180, 2008.

Sian Lewis. Pruning the dendritic tree. *Nature Reviews Neuroscience*, 12(9): 493–493, 2011.

Ashok Litwin-Kumar and Brent Doiron. Formation and maintenance of neuronal assemblies through synaptic plasticity. *Nature communications*, 5(1):5319, 2014.

Cyrille Mascart, Gilles Scarella, Patricia Reynaud-Bouret, and Alexandre Muzy. Scalability of large neural network simulations via activity tracking with time asynchrony and procedural connectivity. *Neural Computation*, 34(9):1915–1943, 2022.

Cyrille Mascart, David Hill, Alexandre Muzy, and Patricia Reynaud-Bouret. Efficient simulation of sparse graphs of point processes. *ACM Transactions on Modeling and Computer Simulation*, 33(1-2):1–27, 2023.

Giulia Mezzadri, Thomas Laloë, Fabien Mathy, and Patricia Reynaud-Bouret. Hold-out strategy for selecting learning models: Application to categorization subjected to presentation orders. *J. Math. Psychol.*, 109:102691, 2022.

Guilherme Ost and Patricia Reynaud-Bouret. Sparse space–time models: Concentration inequalities and lasso. 2020.

Tien Cuong Phi, Alexandre Muzy, and Patricia Reynaud-Bouret. Event-scheduling algorithms with kalikow decomposition for simulating potentially infinite neuronal networks. *SN Comput. Sci.*, 1(1):1–10, 2020.

Patricia Reynaud-Bouret and Sophie Schbath. Adaptive estimation for hawkes processes; application to genome analysis. 2010.

Patricia Reynaud-Bouret, Vincent Rivoirard, and Christine Tuleau-Malot. Inference of functional connectivity in neurosciences via hawkes processes. In *GlobalSIP*, pages 317–320, 2013.

Raanan Y. Rohekar, Yaniv Gurwicz, and Shami Nisimov. Causal interpretation of self-attention in pre-trained transformers. In *NeurIPS*, 2023.

Frank Rosenblatt. *The perceptron, a perceiving and recognizing automaton:(Project Para)*. Cornell Aeronautical Laboratory, 1957.

Frank Rosenblatt et al. *Principles of neurodynamics: Perceptrons and the theory of brain mechanisms*, volume 55. Spartan books Washington, DC, 1962.

Roman Rybka, Yury Davydov, Danila Vlasov, Alexey Serenko, Alexander Sboev, and Vyacheslav Ilyin. Comparison of bagging and sparcity methods for connectivity reduction in spiking neural networks with memristive plasticity. *Big Data and Cognitive Computing*, 8(3):22, 2024.

Alexander Sboev, Yury Davydov, Roman Rybka, Danila Vlasov, and Alexey Serenko. A comparison of two variants of memristive plasticity for solving the classification problem of handwritten digits recognition. In *Biologically Inspired Cognitive Architectures 2021*, pages 438–446, 2022.

Wolfram Schultz. Predictive reward signal of dopamine neurons. *Journal of neurophysiology*, 1998.

Shai Shalev-Shwartz and Shai Ben-David. *Understanding Machine Learning: From Theory to Algorithms*. Cambridge University Press, 2014.

Wolf Singer, Andreas K Engel, Andreas K Kreiter, Matthias HJ Munk, Sergio Neuenschwander, and Pieter R Roelfsema. Neuronal assemblies: necessity, signature and detectability. *Trends in cognitive sciences*, 1(7):252–261, 1997.

James V Stone. Principles of neural information theory. *Computational Neuroscience and Metabolic Efficiency*, 2018.

Amirhossein Tavanaei, Masoud Ghodrati, Saeed Reza Kheradpisheh, Timothée Masquelier, and Anthony Maida. Deep learning in spiking neural networks. *Neural Netw.*, 111:47–63, 2019.

K Türkyilmaz, Maria Nicolette Margaretha van Lieshout, and Alfred Stein. Comparing the hawkes and trigger process models for aftershock sequences following the 2005 kashmir earthquake. *Mathematical geosciences*, 45:149–164, 2013.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. *NeurIPS*, 2017.

Ke Zhou, Hongyuan Zha, and Le Song. Learning social infectivity in sparse low-rank networks using multi-dimensional hawkes processes. In *AISTATS*, 2013.

Simiao Zuo, Haoming Jiang, Zichong Li, Tuo Zhao, and Hongyuan Zha. Transformer Hawkes process. In *ICML*, 2020.